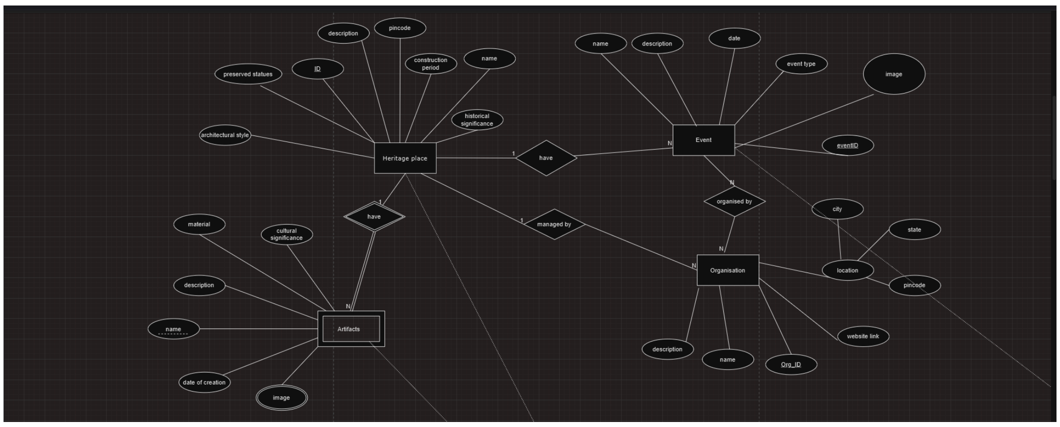
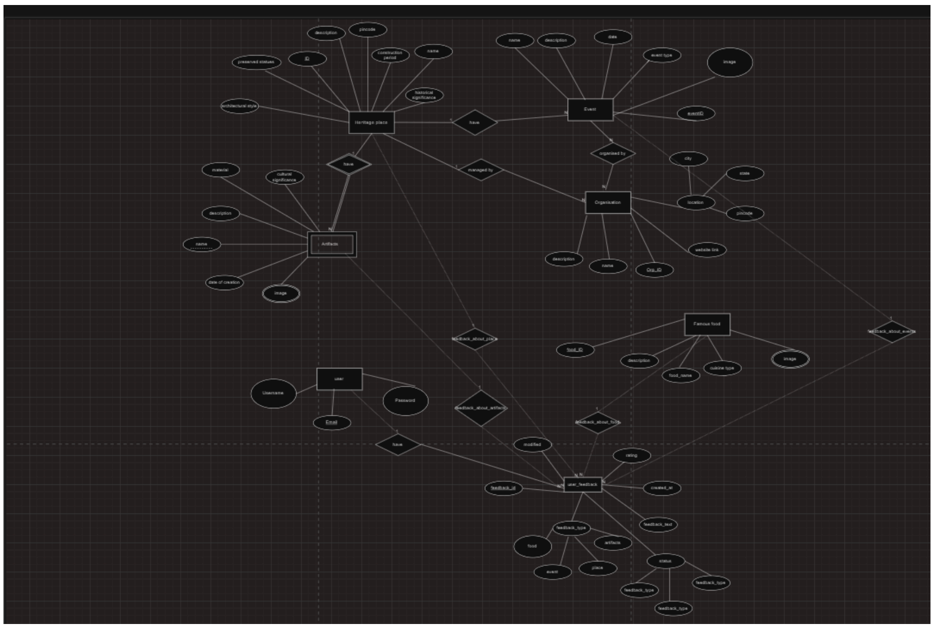
DBMS PROJECT REPORT

Cultural Heritage and Cuisine Archive

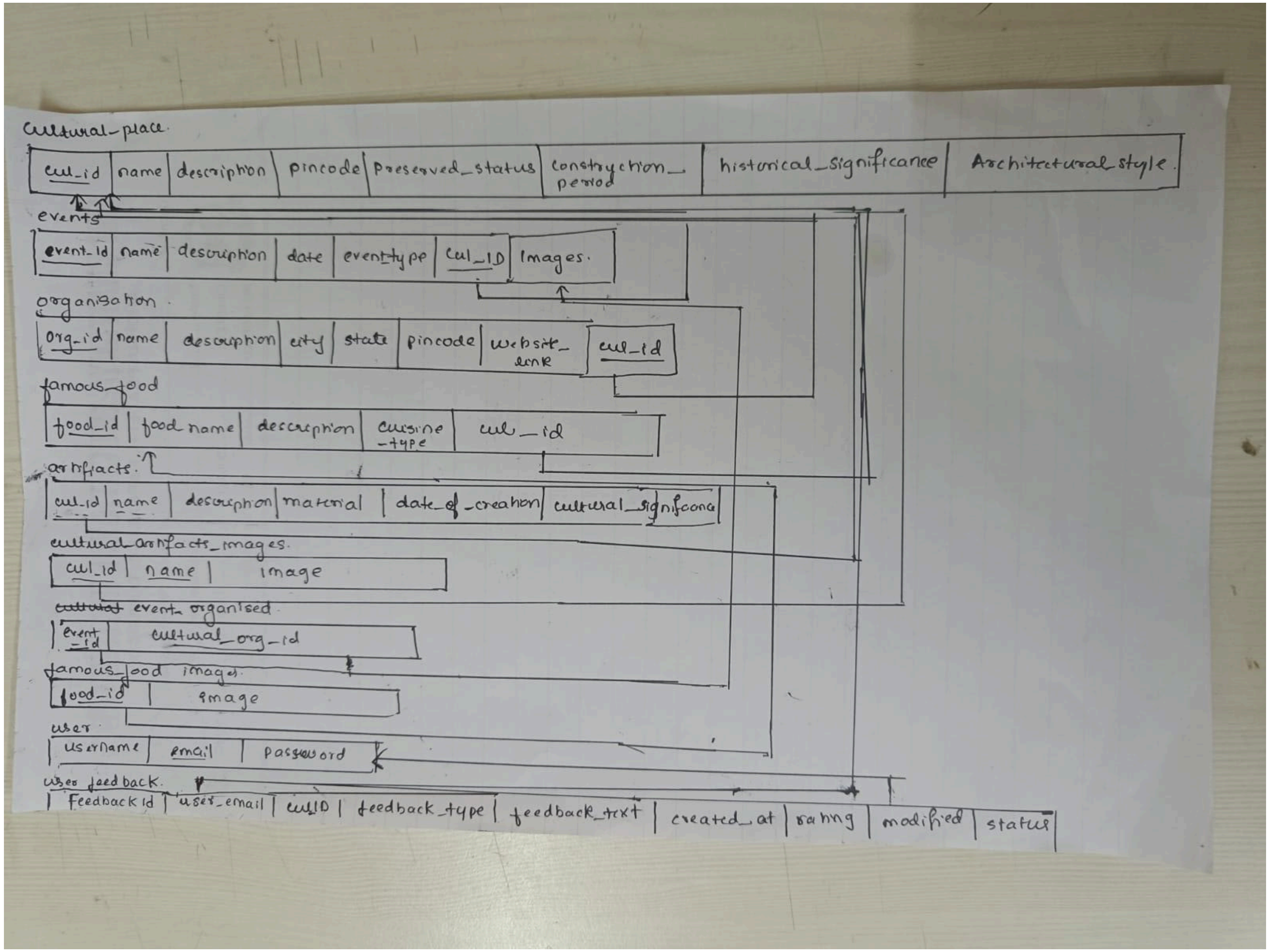
Name: Y M Rashmi	**Name: Vinayashree N. G**
SRN: PES1UG22CS712	**SRN: PES1UG22CS696**

ERDiagram



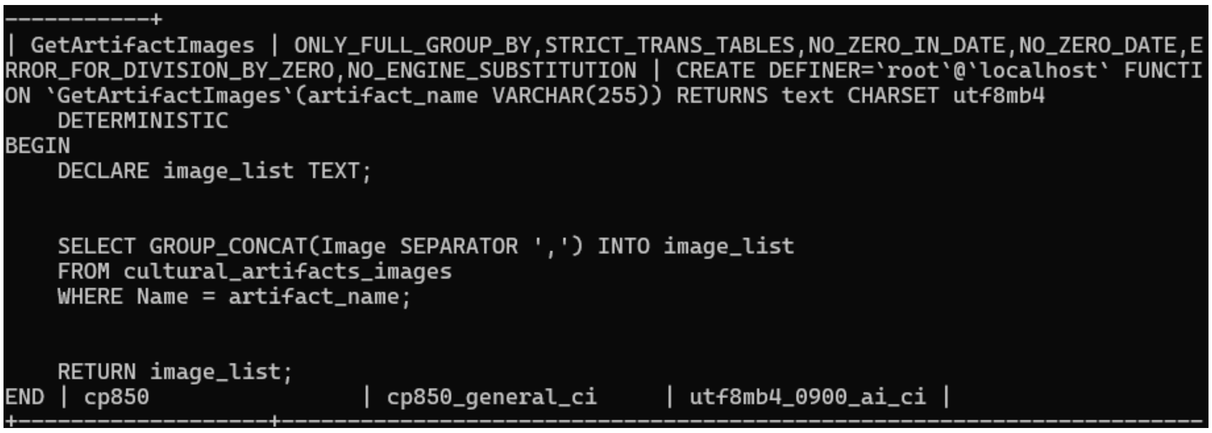


Relational Scheme:


cultural-place.
cul_id | name | description | pincode | Preserved_status | construction_period | historical_significance | Architectural_style.
events
event_id | name | description | date | event_type | cul_ID | Images.
organisation.
org_id | name | description | city | state | pincode | websit_link | cul_id
famous_food
food_id | food name | description | cuisine_type | cul_id
artifacts.
cul_id | name | description | material | date_of_creation | cultural_significance
cultural_artifacts_images.
cul_id | name | image
~~cultural~~ event_organised.
event_id | cultural_org_id
famous_food images.
food_id | image
user
username | email | password
user feedback.
Feedback id | user_email | culID | feedback_type | feedback_text | created_at | rating | modified | status

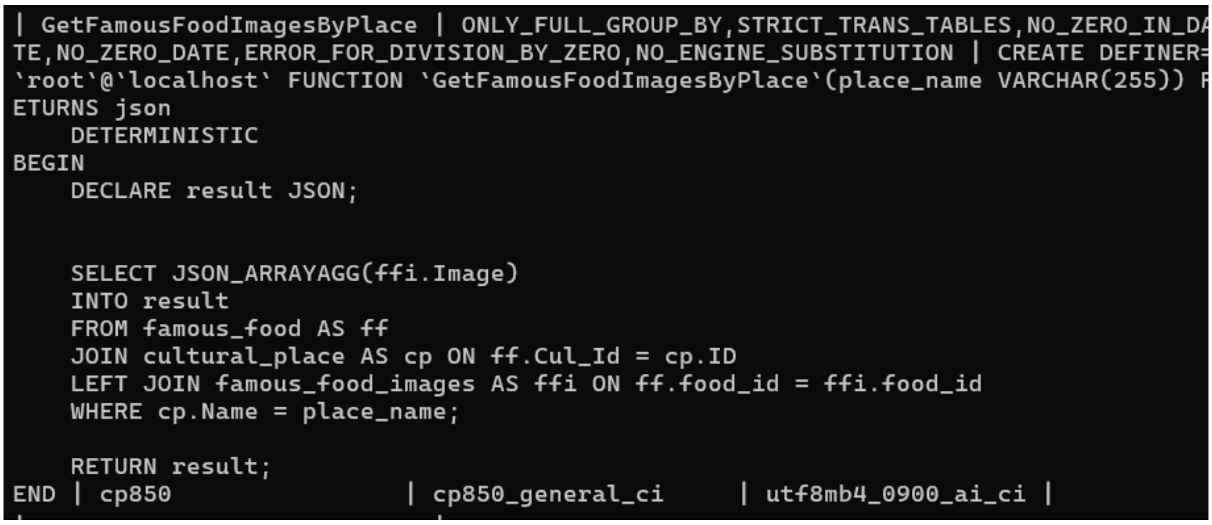
Functions:

1)GetArtifactImage
The functions retrieves and concatenates all image names associated with a specified artifact (`artifact_name`) from the `cultural_artifacts_images` table into a single comma-separated string, which is then returned as `image_list`



```
-----------+
| GetArtifactImages | ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,E
RROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION | CREATE DEFINER=`root`@`localhost` FUNCTI
ON `GetArtifactImages`(artifact_name VARCHAR(255)) RETURNS text CHARSET utf8mb4
    DETERMINISTIC
BEGIN
    DECLARE image_list TEXT;


    SELECT GROUP_CONCAT(Image SEPARATOR ',') INTO image_list
    FROM cultural_artifacts_images
    WHERE Name = artifact_name;


    RETURN image_list;
END | cp850                | cp850_general_ci     | utf8mb4_0900_ai_ci |
+--------------------+--------------------------------------------------------------------
```

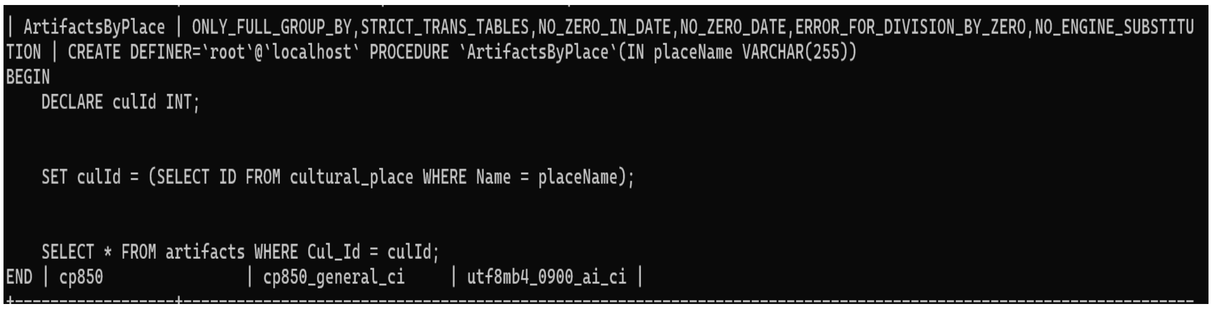
2)GetFamousFoodImagesByPlace
This function retrieves all images of famous foods associated with a specified cultural place (`place_name`) by joining `famous_food`, `cultural_place`, and `famous_food_images` tables, and returns the result as a JSON array.



```
| GetFamousFoodImagesByPlace | ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DA
TE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION | CREATE DEFINER=
`root`@`localhost` FUNCTION `GetFamousFoodImagesByPlace`(place_name VARCHAR(255)) R
ETURNS json
    DETERMINISTIC
BEGIN
    DECLARE result JSON;


    SELECT JSON_ARRAYAGG(ffi.Image)
    INTO result
    FROM famous_food AS ff
    JOIN cultural_place AS cp ON ff.Cul_Id = cp.ID
    LEFT JOIN famous_food_images AS ffi ON ff.food_id = ffi.food_id
    WHERE cp.Name = place_name;

    RETURN result;
END | cp850                | cp850_general_ci     | utf8mb4_0900_ai_ci |
```

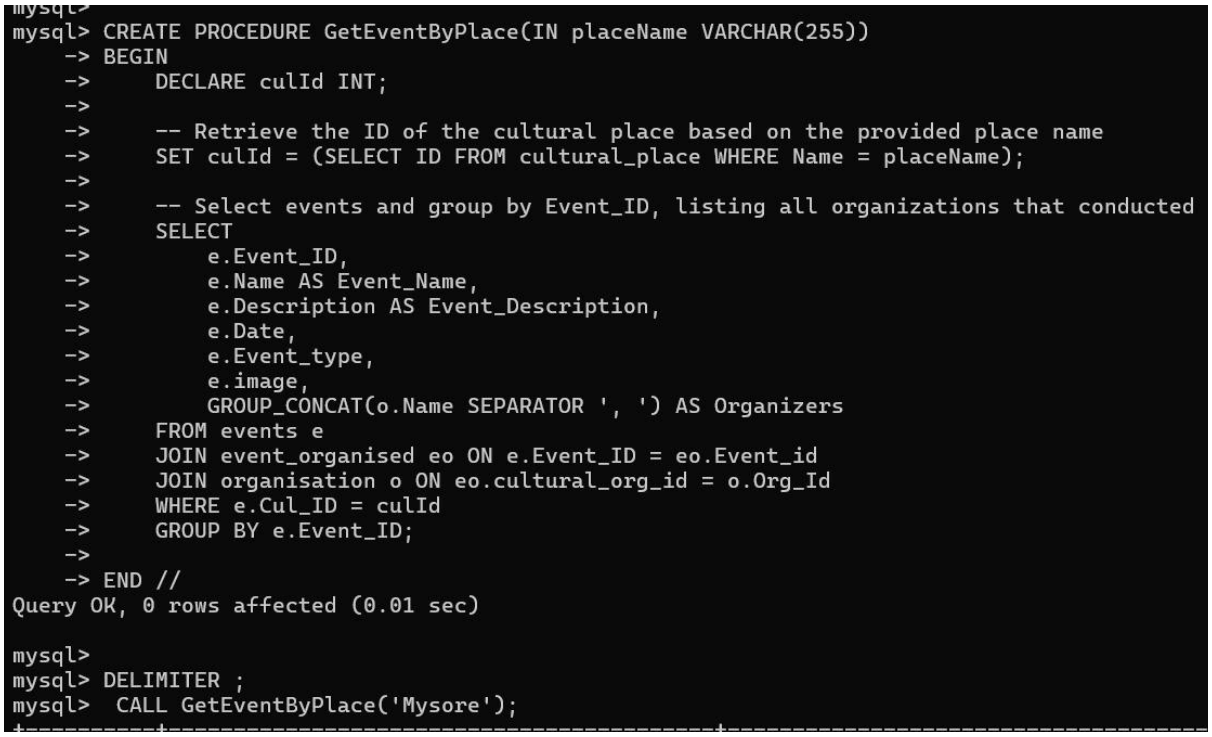
Procedures:

1)ArtifactsByPlace
This procedure retrieves all artifacts associated with a specified cultural place (`placeName`) by first finding its `Cul_Id` from the `cultural_place` table and then selecting matching entries from the `artifacts` table.



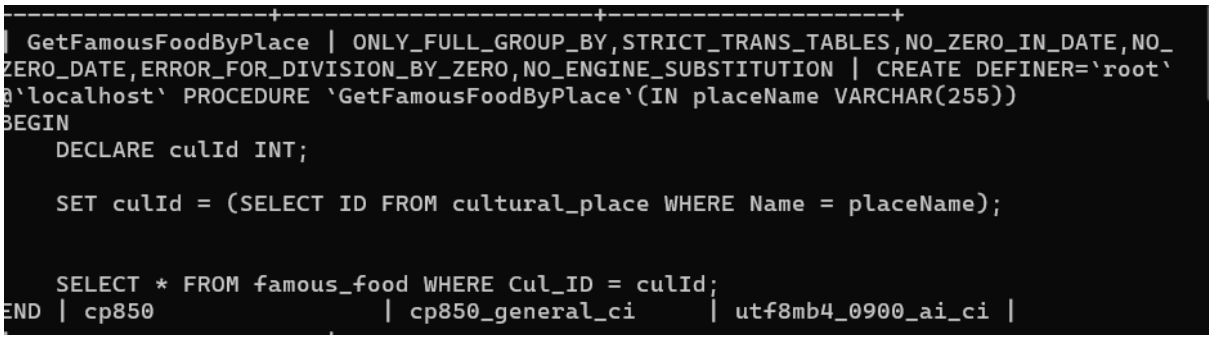
```
| ArtifactsByPlace | ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITU
TION | CREATE DEFINER=`root`@`localhost` PROCEDURE `ArtifactsByPlace`(IN placeName VARCHAR(255))
BEGIN
    DECLARE culId INT;


    SET culId = (SELECT ID FROM cultural_place WHERE Name = placeName);


    SELECT * FROM artifacts WHERE Cul_Id = culId;
END | cp850                | cp850_general_ci     | utf8mb4_0900_ai_ci |
+-----------------+------------------------------------------------------------------------------------------------------
```

2)`GetEventByPlace`
This procedure retrieves all events linked to a specific cultural place by its name. It fetches event details (name, description, date, type, image) and lists the organizers associated with each event, grouping results by `Event_ID`.



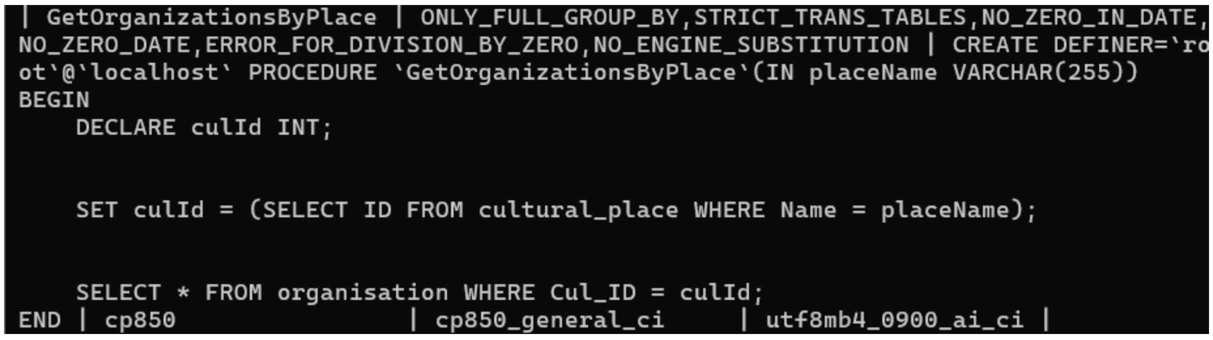
```
mysql>
mysql> CREATE PROCEDURE GetEventByPlace(IN placeName VARCHAR(255))
    -> BEGIN
    ->     DECLARE culId INT;
    ->
    ->     -- Retrieve the ID of the cultural place based on the provided place name
    ->     SET culId = (SELECT ID FROM cultural_place WHERE Name = placeName);
    ->
    ->     -- Select events and group by Event_ID, listing all organizations that conducted
    ->     SELECT
    ->         e.Event_ID,
    ->         e.Name AS Event_Name,
    ->         e.Description AS Event_Description,
    ->         e.Date,
    ->         e.Event_type,
    ->         e.image,
    ->         GROUP_CONCAT(o.Name SEPARATOR ', ') AS Organizers
    ->     FROM events e
    ->     JOIN event_organised eo ON e.Event_ID = eo.Event_id
    ->     JOIN organisation o ON eo.cultural_org_id = o.Org_Id
    ->     WHERE e.Cul_ID = culId
    ->     GROUP BY e.Event_ID;
    ->
    -> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql>  CALL GetEventByPlace('Mysore');
+----------+------------------------------------------+------------------------------------
```

3) GetFamousFoodByPlace
This procedure retrieves all famous foods associated with a specified cultural place (`placeName`) by first finding its `Cul_Id` from the `cultural_place` table and then selecting matching entries from the `famous_food` table.



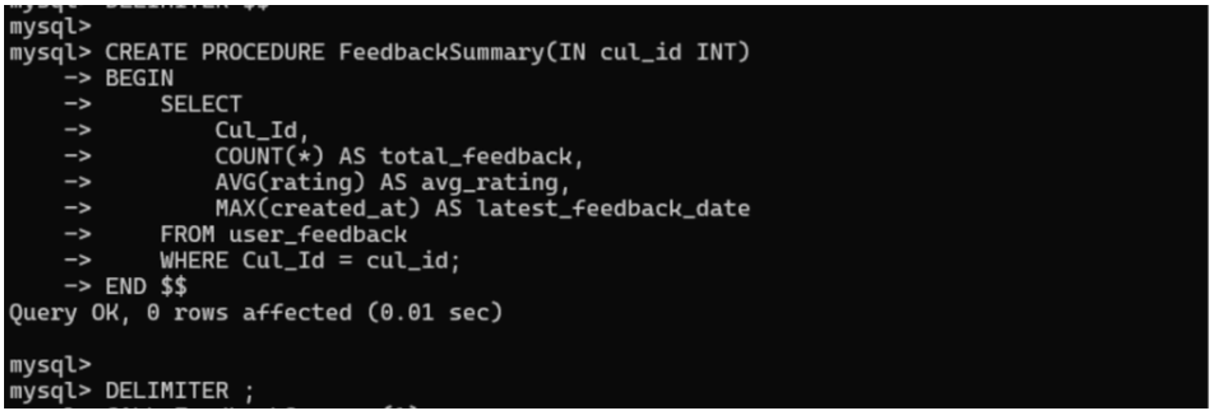
```
-------------------+----------------------+--------------------+
| GetFamousFoodByPlace | ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_
ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION | CREATE DEFINER=`root`
@`localhost` PROCEDURE `GetFamousFoodByPlace`(IN placeName VARCHAR(255))
BEGIN
    DECLARE culId INT;

    SET culId = (SELECT ID FROM cultural_place WHERE Name = placeName);


    SELECT * FROM famous_food WHERE Cul_ID = culId;
END | cp850                 | cp850_general_ci     | utf8mb4_0900_ai_ci |
```

4) GetOrganisationByPlace
This procedure retrieves all organizations associated with a specified cultural place (`placeName`) by first finding its `Cul_Id` from the `cultural_place` table and then selecting matching entries from the `organisation` table.

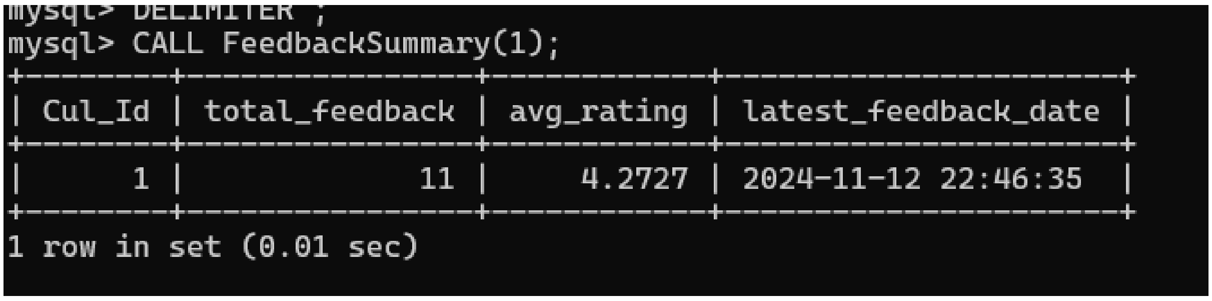


```
| GetOrganizationsByPlace | ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,
NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION | CREATE DEFINER=`ro
ot`@`localhost` PROCEDURE `GetOrganizationsByPlace`(IN placeName VARCHAR(255))
BEGIN
    DECLARE culId INT;


    SET culId = (SELECT ID FROM cultural_place WHERE Name = placeName);


    SELECT * FROM organisation WHERE Cul_ID = culId;
END | cp850                 | cp850_general_ci     | utf8mb4_0900_ai_ci |
```

5)FeedBackSummary
This procedure provides a summary of feedback for a given cultural item, including the number of feedback entries, average rating, and latest feedback text.

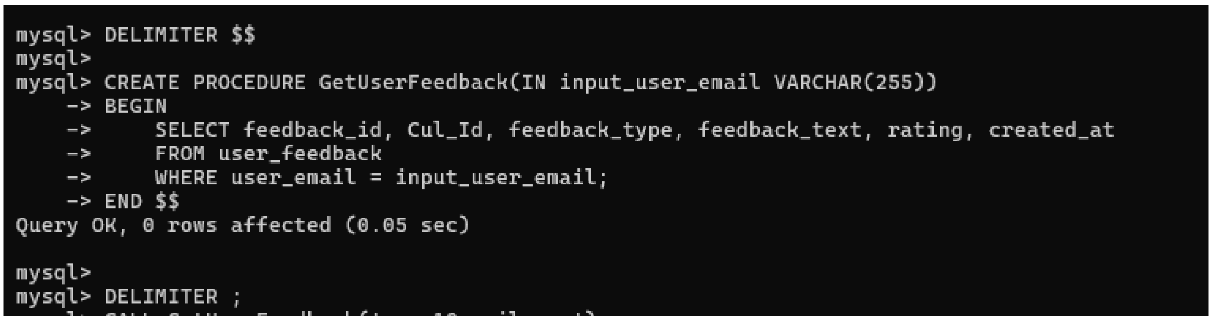


```
mysql>
mysql> CREATE PROCEDURE FeedbackSummary(IN cul_id INT)
    -> BEGIN
    ->     SELECT
    ->         Cul_Id,
    ->         COUNT(*) AS total_feedback,
    ->         AVG(rating) AS avg_rating,
    ->         MAX(created_at) AS latest_feedback_date
    ->     FROM user_feedback
    ->     WHERE Cul_Id = cul_id;
    -> END $$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
```

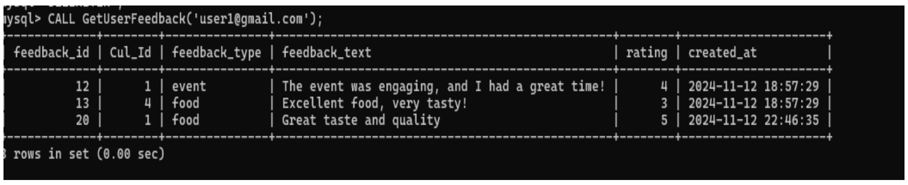


```
mysql> DELIMITER ;
mysql> CALL FeedbackSummary(1);
+--------+----------------+------------+---------------------+
| Cul_Id | total_feedback | avg_rating | latest_feedback_date |
+--------+----------------+------------+---------------------+
|      1 |             11 |     4.2727 | 2024-11-12 22:46:35  |
+--------+----------------+------------+---------------------+
1 row in set (0.01 sec)
```

6)GetUserFeedback
This procedure retrieves all feedback entries by a specified user, which can help in building a user feedback history.



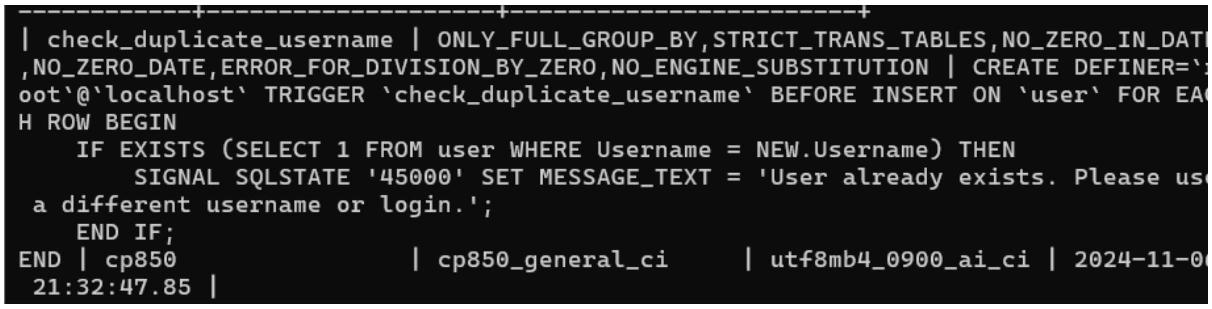
```
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetUserFeedback(IN input_user_email VARCHAR(255))
    -> BEGIN
    ->     SELECT feedback_id, Cul_Id, feedback_type, feedback_text, rating, created_at
    ->     FROM user_feedback
    ->     WHERE user_email = input_user_email;
    -> END $$
Query OK, 0 rows affected (0.05 sec)

mysql>
mysql> DELIMITER ;
```



```
ysql> CALL GetUserFeedback('user1@gmail.com');
+-------------+--------+---------------+------------------------------------------------+--------+---------------------+
| feedback_id | Cul_Id | feedback_type | feedback_text                                  | rating | created_at          |
+-------------+--------+---------------+------------------------------------------------+--------+---------------------+
|          12 |      1 | event         | The event was engaging, and I had a great time! |      4 | 2024-11-12 18:57:29 |
|          13 |      4 | food          | Excellent food, very tasty!                    |      3 | 2024-11-12 18:57:29 |
|          20 |      1 | food          | Great taste and quality                        |      5 | 2024-11-12 22:46:35 |
+-------------+--------+---------------+------------------------------------------------+--------+---------------------+
 rows in set (0.00 sec)
```

Triggers:

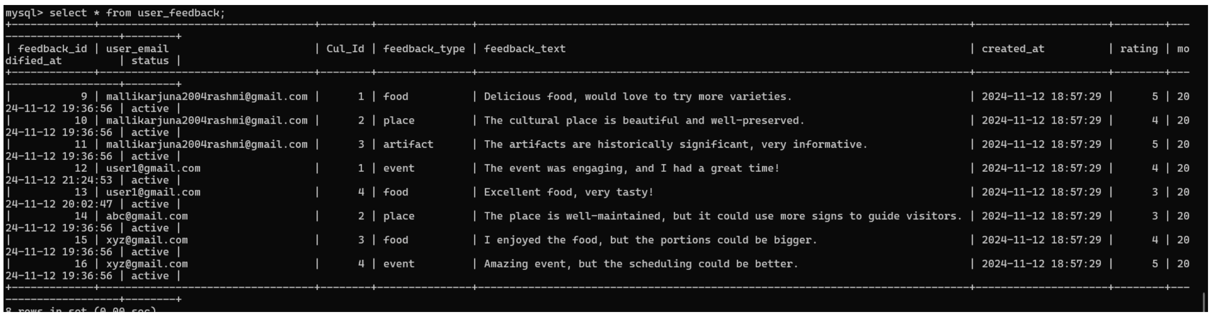
1. `Check_duplicate_username`,
This trigger ensures uniqueness of usernames in the `user` table. Before inserting a new record, it checks if the `Username` already exists; if so, it raises an error with the message "User already exists. Please use a different username or login."



```
------------+--------------------+-----------------------+
| check_duplicate_username | ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DAT
,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION | CREATE DEFINER=`
oot`@`localhost` TRIGGER `check_duplicate_username` BEFORE INSERT ON `user` FOR EA
H ROW BEGIN
    IF EXISTS (SELECT 1 FROM user WHERE Username = NEW.Username) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'User already exists. Please us
 a different username or login.';
    END IF;
END | cp850                | cp850_general_ci     | utf8mb4_0900_ai_ci | 2024-11-0
 21:32:47.85 |
```

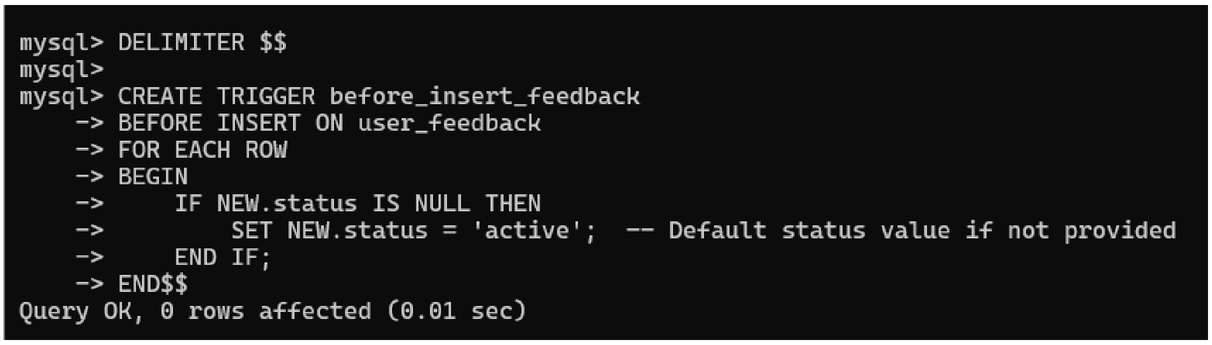
2) Before_insert_feedback
This trigger is used to set the `status` field to `'active'` by default before inserting a new feedback record.

Before trigger Execution:



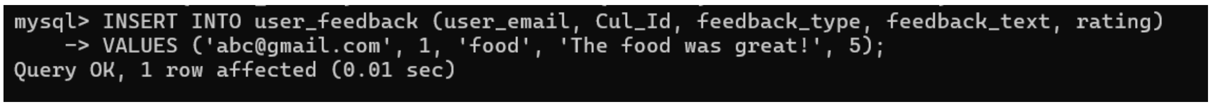
```
mysql> select * from user_feedback;
+-------------+-----------------------------------+--------+---------------+----------------------------------------------------------------------------+---------------------+--------+---
------------------+--------+
| feedback_id | user_email                        | Cul_Id | feedback_type | feedback_text                                                              | created_at          | rating | mo
dified_at         | status |
+-------------+-----------------------------------+--------+---------------+----------------------------------------------------------------------------+---------------------+--------+---
------------------+--------+
|           9 | mallikarjuna2004rashmi@gmail.com  |      1 | food          | Delicious food, would love to try more varieties.                          | 2024-11-12 18:57:29 |      5 | 20
24-11-12 19:36:56 | active |
|          10 | mallikarjuna2004rashmi@gmail.com  |      2 | place         | The cultural place is beautiful and well-preserved.                        | 2024-11-12 18:57:29 |      4 | 20
24-11-12 19:36:56 | active |
|          11 | mallikarjuna2004rashmi@gmail.com  |      3 | artifact      | The artifacts are historically significant, very informative.              | 2024-11-12 18:57:29 |      5 | 20
24-11-12 19:36:56 | active |
|          12 | user1@gmail.com                   |      1 | event         | The event was engaging, and I had a great time!                            | 2024-11-12 18:57:29 |      4 | 20
24-11-12 21:24:53 | active |
|          13 | user1@gmail.com                   |      4 | food          | Excellent food, very tasty!                                                | 2024-11-12 18:57:29 |      3 | 20
24-11-12 20:02:47 | active |
|          14 | abc@gmail.com                     |      2 | place         | The place is well-maintained, but it could use more signs to guide visitors. | 2024-11-12 18:57:29 |      3 | 20
24-11-12 19:36:56 | active |
|          15 | xyz@gmail.com                     |      3 | food          | I enjoyed the food, but the portions could be bigger.                      | 2024-11-12 18:57:29 |      4 | 20
24-11-12 19:36:56 | active |
|          16 | xyz@gmail.com                     |      4 | event         | Amazing event, but the scheduling could be better.                         | 2024-11-12 18:57:29 |      5 | 20
24-11-12 19:36:56 | active |
+-------------+-----------------------------------+--------+---------------+----------------------------------------------------------------------------+---------------------+--------+---
------------------+--------+
8 rows in set (0.00 sec)
```

Trigger code:

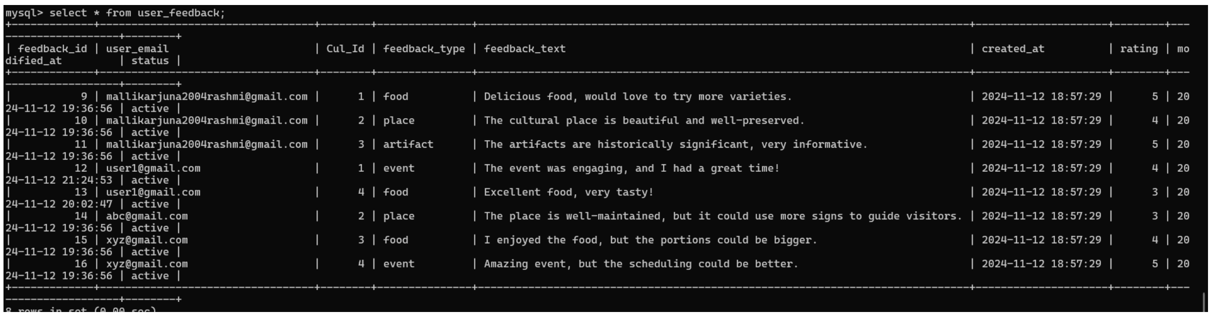


```
mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER before_insert_feedback
    -> BEFORE INSERT ON user_feedback
    -> FOR EACH ROW
    -> BEGIN
    ->     IF NEW.status IS NULL THEN
    ->         SET NEW.status = 'active';  -- Default status value if not provided
    ->     END IF;
    -> END$$
Query OK, 0 rows affected (0.01 sec)
```

After Trigger Execution:



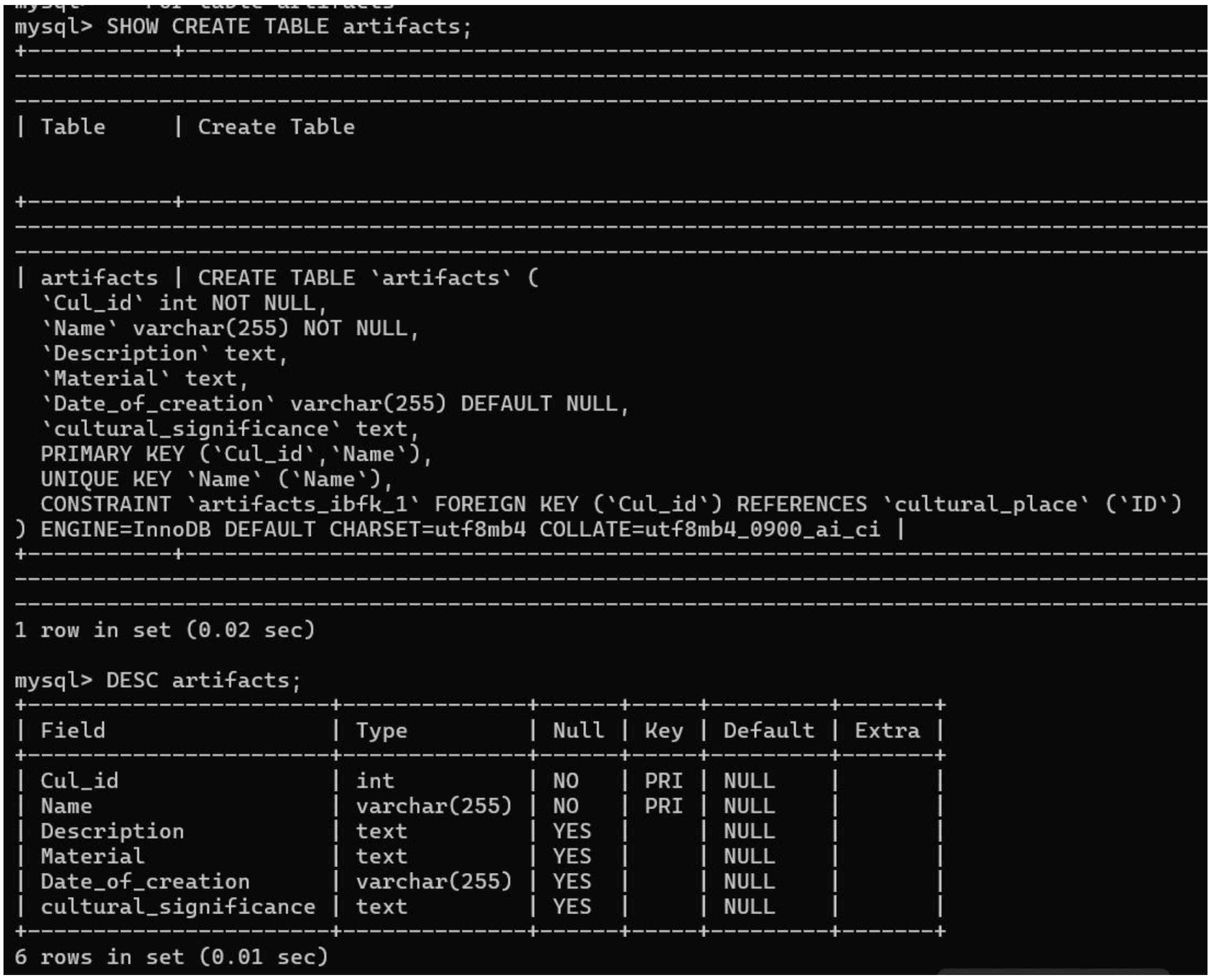
```
mysql> INSERT INTO user_feedback (user_email, Cul_Id, feedback_type, feedback_text, rating)
    -> VALUES ('abc@gmail.com', 1, 'food', 'The food was great!', 5);
Query OK, 1 row affected (0.01 sec)
```



```
mysql> select * from user_feedback;
+-------------+-----------------------------------+--------+---------------+----------------------------------------------------------------------------+---------------------+--------+---
------------------+--------+
| feedback_id | user_email                        | Cul_Id | feedback_type | feedback_text                                                              | created_at          | rating | mo
dified_at         | status |
+-------------+-----------------------------------+--------+---------------+----------------------------------------------------------------------------+---------------------+--------+---
------------------+--------+
|           9 | mallikarjuna2004rashmi@gmail.com  |      1 | food          | Delicious food, would love to try more varieties.                          | 2024-11-12 18:57:29 |      5 | 20
24-11-12 19:36:56 | active |
|          10 | mallikarjuna2004rashmi@gmail.com  |      2 | place         | The cultural place is beautiful and well-preserved.                        | 2024-11-12 18:57:29 |      4 | 20
24-11-12 19:36:56 | active |
|          11 | mallikarjuna2004rashmi@gmail.com  |      3 | artifact      | The artifacts are historically significant, very informative.              | 2024-11-12 18:57:29 |      5 | 20
24-11-12 19:36:56 | active |
|          12 | user1@gmail.com                   |      1 | event         | The event was engaging, and I had a great time!                            | 2024-11-12 18:57:29 |      4 | 20
24-11-12 21:24:53 | active |
|          13 | user1@gmail.com                   |      4 | food          | Excellent food, very tasty!                                                | 2024-11-12 18:57:29 |      3 | 20
24-11-12 20:02:47 | active |
|          14 | abc@gmail.com                     |      2 | place         | The place is well-maintained, but it could use more signs to guide visitors. | 2024-11-12 18:57:29 |      3 | 20
24-11-12 19:36:56 | active |
|          15 | xyz@gmail.com                     |      3 | food          | I enjoyed the food, but the portions could be bigger.                      | 2024-11-12 18:57:29 |      4 | 20
24-11-12 19:36:56 | active |
|          16 | xyz@gmail.com                     |      4 | event         | Amazing event, but the scheduling could be better.                         | 2024-11-12 18:57:29 |      5 | 20
24-11-12 19:36:56 | active |
+-------------+-----------------------------------+--------+---------------+----------------------------------------------------------------------------+---------------------+--------+---
------------------+--------+
8 rows in set (0.00 sec)
```

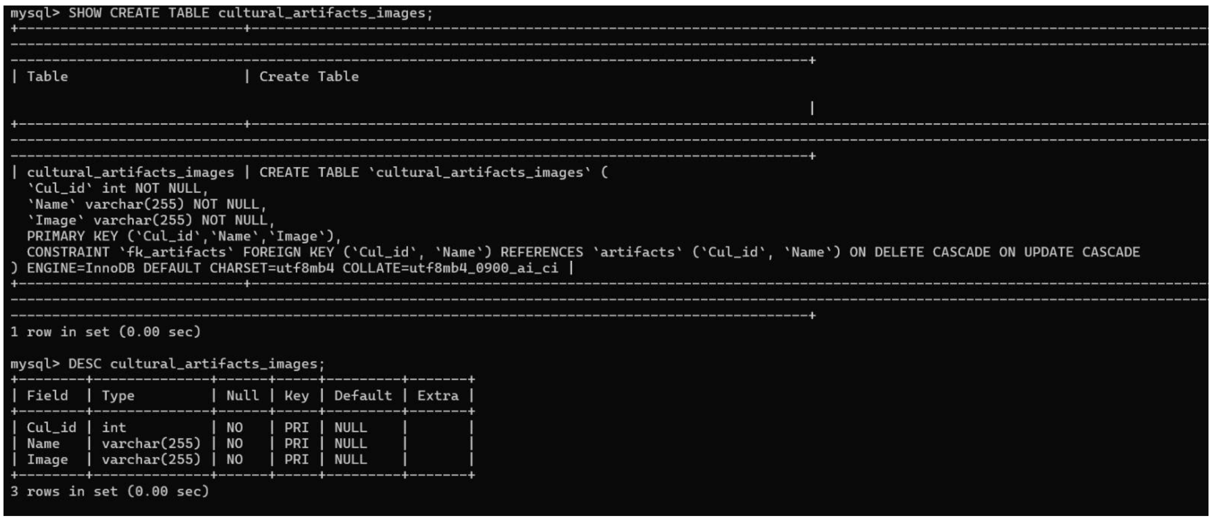
Tables

Artifact table:



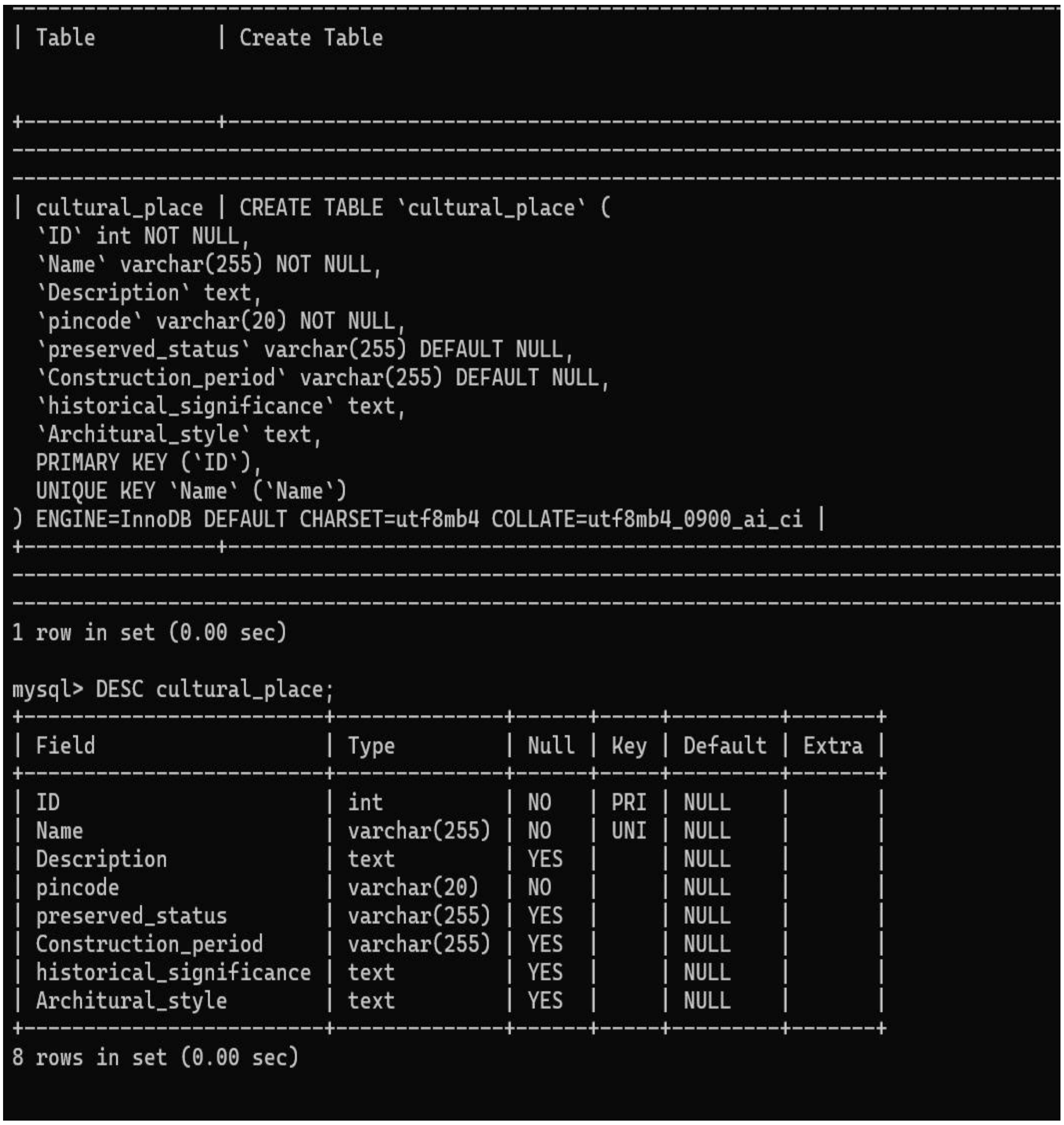
```
mysql> SHOW CREATE TABLE artifacts;
+-----------+--------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
| Table     | Create Table


+-----------+--------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
| artifacts | CREATE TABLE `artifacts` (
  `Cul_id` int NOT NULL,
  `Name` varchar(255) NOT NULL,
  `Description` text,
  `Material` text,
  `Date_of_creation` varchar(255) DEFAULT NULL,
  `cultural_significance` text,
  PRIMARY KEY (`Cul_id`,`Name`),
  UNIQUE KEY `Name` (`Name`),
  CONSTRAINT `artifacts_ibfk_1` FOREIGN KEY (`Cul_id`) REFERENCES `cultural_place` (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----------+--------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
1 row in set (0.02 sec)

mysql> DESC artifacts;
+-----------------------+--------------+------+-----+---------+-------+
| Field                 | Type         | Null | Key | Default | Extra |
+-----------------------+--------------+------+-----+---------+-------+
| Cul_id                | int          | NO   | PRI | NULL    |       |
| Name                  | varchar(255) | NO   | PRI | NULL    |       |
| Description           | text         | YES  |     | NULL    |       |
| Material              | text         | YES  |     | NULL    |       |
| Date_of_creation      | varchar(255) | YES  |     | NULL    |       |
| cultural_significance | text         | YES  |     | NULL    |       |
+-----------------------+--------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

Artifact Image Table:



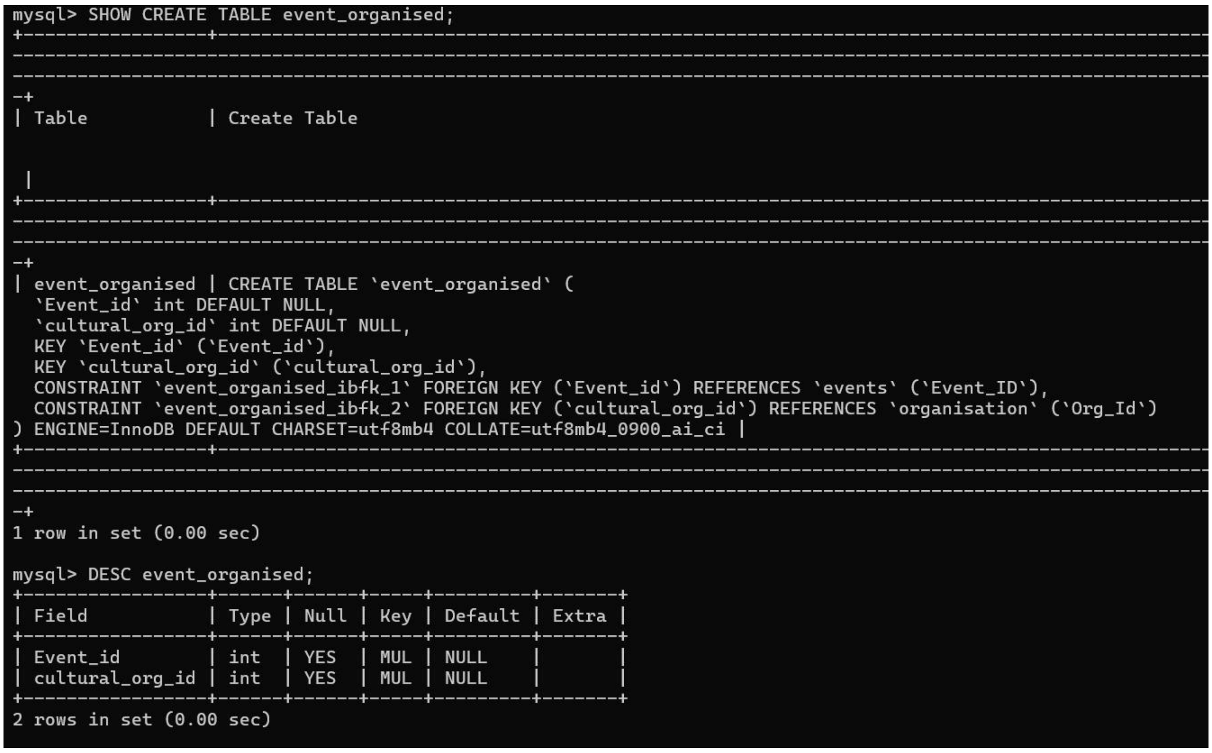
```
mysql> SHOW CREATE TABLE cultural_artifacts_images;
+---------------------------+--------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------+
| Table                     | Create Table

                                                                                                  |
+---------------------------+--------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------+
| cultural_artifacts_images | CREATE TABLE `cultural_artifacts_images` (
  `Cul_id` int NOT NULL,
  `Name` varchar(255) NOT NULL,
  `Image` varchar(255) NOT NULL,
  PRIMARY KEY (`Cul_id`,`Name`,`Image`),
  CONSTRAINT `fk_artifacts` FOREIGN KEY (`Cul_id`, `Name`) REFERENCES `artifacts` (`Cul_id`, `Name`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+---------------------------+--------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)

mysql> DESC cultural_artifacts_images;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| Cul_id | int          | NO   | PRI | NULL    |       |
| Name   | varchar(255) | NO   | PRI | NULL    |       |
| Image  | varchar(255) | NO   | PRI | NULL    |       |
+--------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

Cultural Place Table



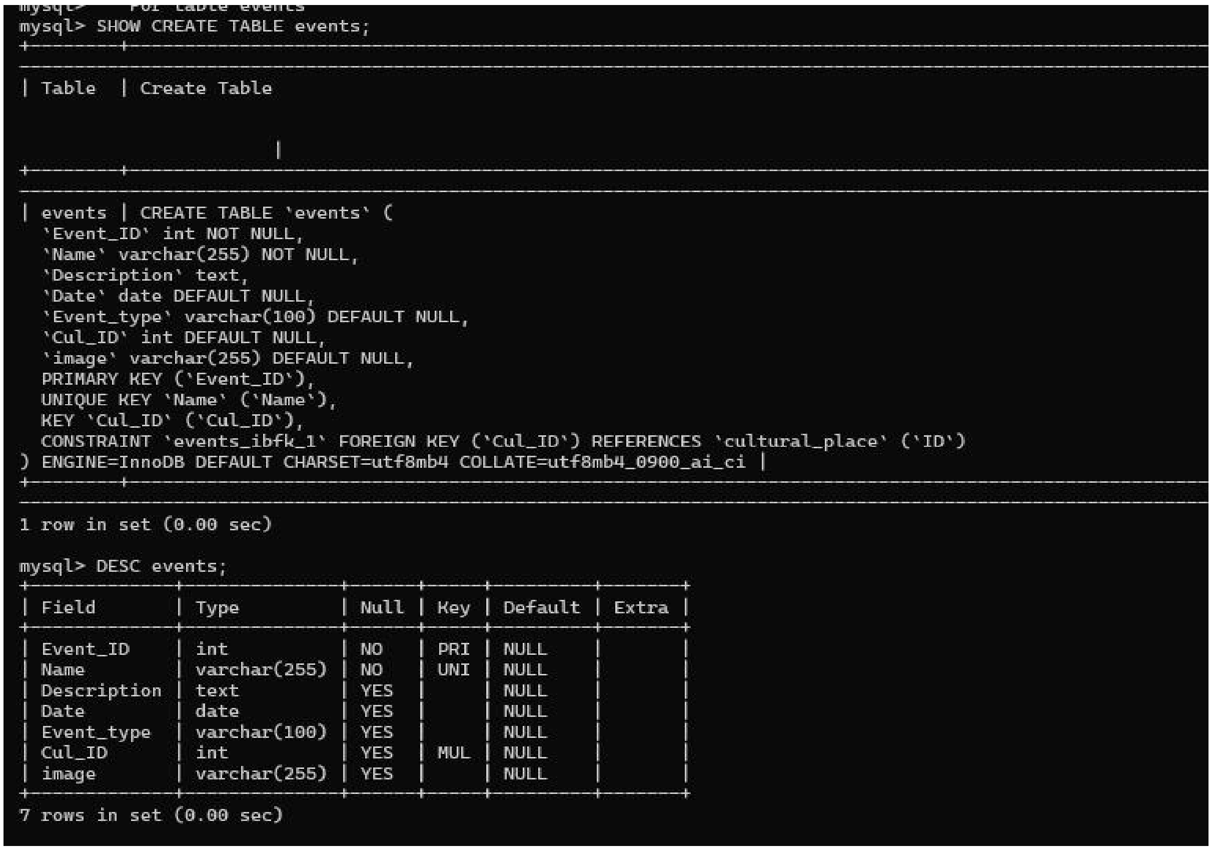
```
| Table          | Create Table


+----------------+-------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
| cultural_place | CREATE TABLE `cultural_place` (
  `ID` int NOT NULL,
  `Name` varchar(255) NOT NULL,
  `Description` text,
  `pincode` varchar(20) NOT NULL,
  `preserved_status` varchar(255) DEFAULT NULL,
  `Construction_period` varchar(255) DEFAULT NULL,
  `historical_significance` text,
  `Architural_style` text,
  PRIMARY KEY (`ID`),
  UNIQUE KEY `Name` (`Name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+----------------+-------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
1 row in set (0.00 sec)

mysql> DESC cultural_place;
+-------------------------+--------------+------+-----+---------+-------+
| Field                   | Type         | Null | Key | Default | Extra |
+-------------------------+--------------+------+-----+---------+-------+
| ID                      | int          | NO   | PRI | NULL    |       |
| Name                    | varchar(255) | NO   | UNI | NULL    |       |
| Description             | text         | YES  |     | NULL    |       |
| pincode                 | varchar(20)  | NO   |     | NULL    |       |
| preserved_status        | varchar(255) | YES  |     | NULL    |       |
| Construction_period     | varchar(255) | YES  |     | NULL    |       |
| historical_significance | text         | YES  |     | NULL    |       |
| Architural_style        | text         | YES  |     | NULL    |       |
+-------------------------+--------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

Event Organised Table



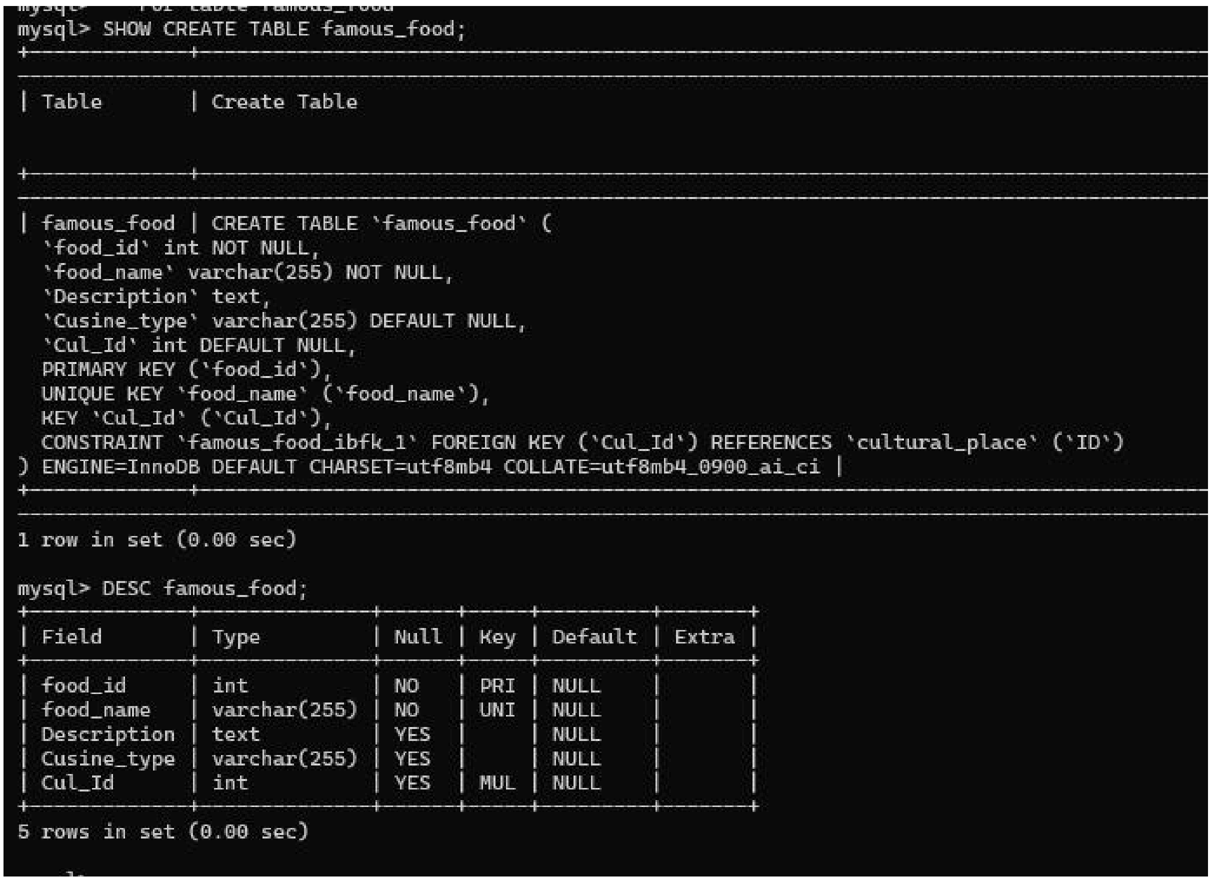
```
mysql> SHOW CREATE TABLE event_organised;
+-----------------+------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
-+
| Table           | Create Table


 |
+-----------------+------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
-+
| event_organised | CREATE TABLE `event_organised` (
  `Event_id` int DEFAULT NULL,
  `cultural_org_id` int DEFAULT NULL,
  KEY `Event_id` (`Event_id`),
  KEY `cultural_org_id` (`cultural_org_id`),
  CONSTRAINT `event_organised_ibfk_1` FOREIGN KEY (`Event_id`) REFERENCES `events` (`Event_ID`),
  CONSTRAINT `event_organised_ibfk_2` FOREIGN KEY (`cultural_org_id`) REFERENCES `organisation` (`Org_Id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----------------+------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
-+
1 row in set (0.00 sec)

mysql> DESC event_organised;
+-----------------+------+------+-----+---------+-------+
| Field           | Type | Null | Key | Default | Extra |
+-----------------+------+------+-----+---------+-------+
| Event_id        | int  | YES  | MUL | NULL    |       |
| cultural_org_id | int  | YES  | MUL | NULL    |       |
+-----------------+------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Events Table



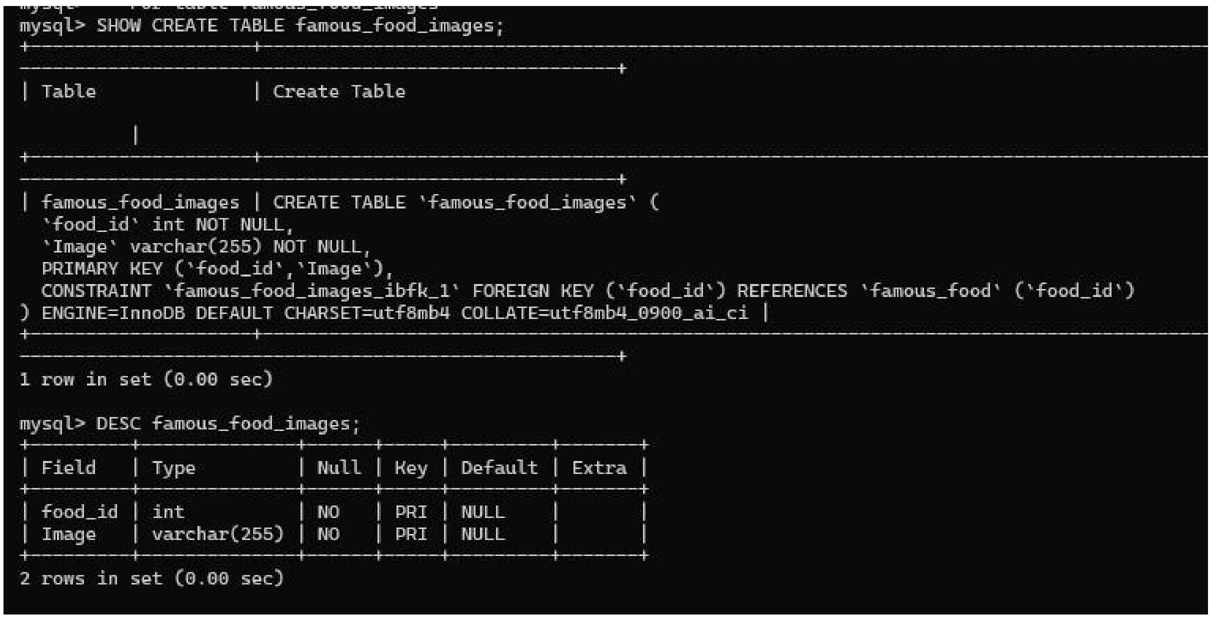
```
mysql>    For table events
mysql> SHOW CREATE TABLE events;
+--------+-------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
| Table  | Create Table


                        |
+--------+-------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
| events | CREATE TABLE `events` (
  `Event_ID` int NOT NULL,
  `Name` varchar(255) NOT NULL,
  `Description` text,
  `Date` date DEFAULT NULL,
  `Event_type` varchar(100) DEFAULT NULL,
  `Cul_ID` int DEFAULT NULL,
  `image` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`Event_ID`),
  UNIQUE KEY `Name` (`Name`),
  KEY `Cul_ID` (`Cul_ID`),
  CONSTRAINT `events_ibfk_1` FOREIGN KEY (`Cul_ID`) REFERENCES `cultural_place` (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+--------+-------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
1 row in set (0.00 sec)

mysql> DESC events;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| Event_ID    | int          | NO   | PRI | NULL    |       |
| Name        | varchar(255) | NO   | UNI | NULL    |       |
| Description | text         | YES  |     | NULL    |       |
| Date        | date         | YES  |     | NULL    |       |
| Event_type  | varchar(100) | YES  |     | NULL    |       |
| Cul_ID      | int          | YES  | MUL | NULL    |       |
| image       | varchar(255) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

Famous Food Table



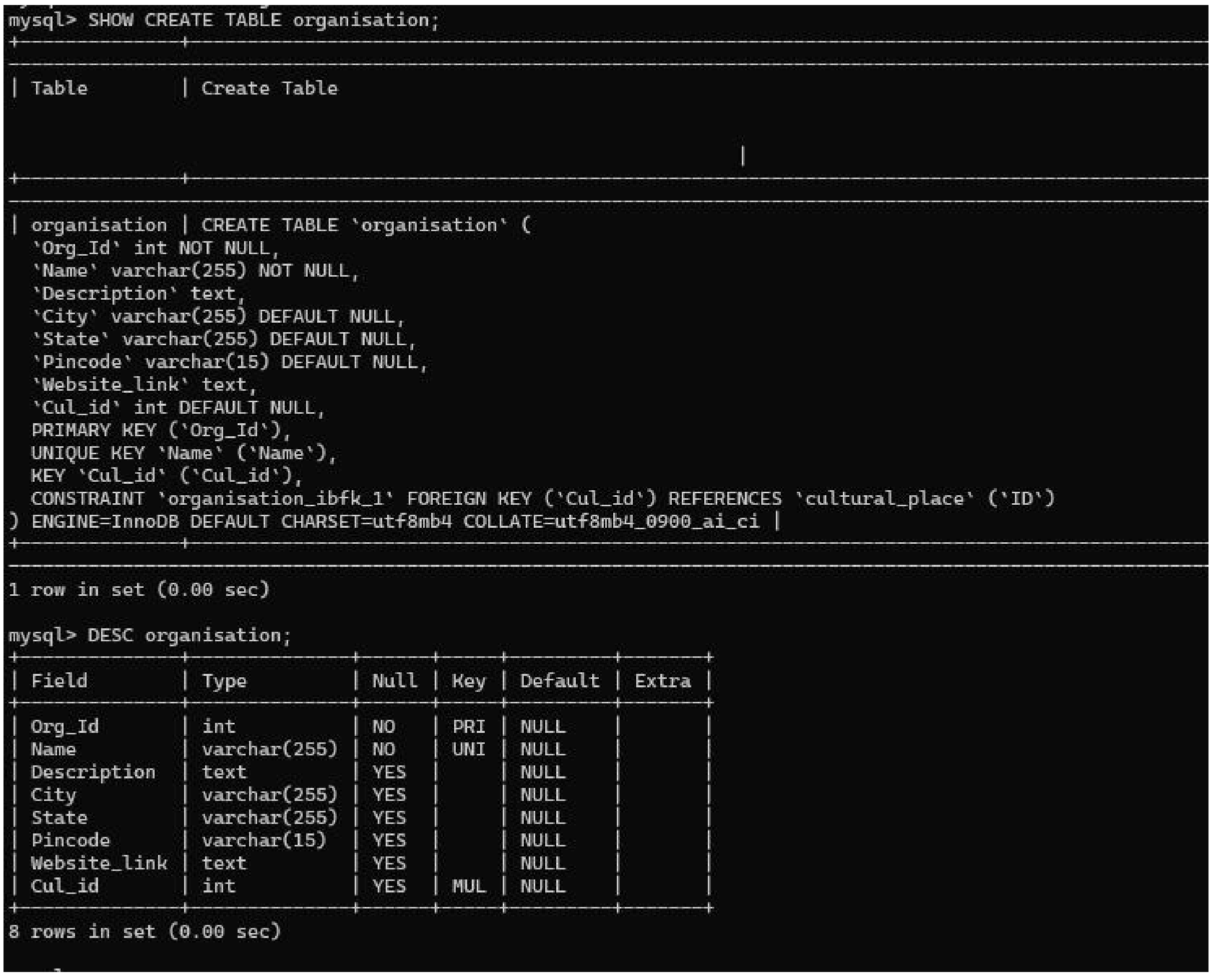
```
mysql> SHOW CREATE TABLE famous_food;
+-------------+---------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------
| Table       | Create Table


+-------------+---------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------
| famous_food | CREATE TABLE `famous_food` (
  `food_id` int NOT NULL,
  `food_name` varchar(255) NOT NULL,
  `Description` text,
  `Cusine_type` varchar(255) DEFAULT NULL,
  `Cul_Id` int DEFAULT NULL,
  PRIMARY KEY (`food_id`),
  UNIQUE KEY `food_name` (`food_name`),
  KEY `Cul_Id` (`Cul_Id`),
  CONSTRAINT `famous_food_ibfk_1` FOREIGN KEY (`Cul_Id`) REFERENCES `cultural_place` (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-------------+---------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------
1 row in set (0.00 sec)

mysql> DESC famous_food;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| food_id     | int          | NO   | PRI | NULL    |       |
| food_name   | varchar(255) | NO   | UNI | NULL    |       |
| Description | text         | YES  |     | NULL    |       |
| Cusine_type | varchar(255) | YES  |     | NULL    |       |
| Cul_Id      | int          | YES  | MUL | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

Famous Food images Table



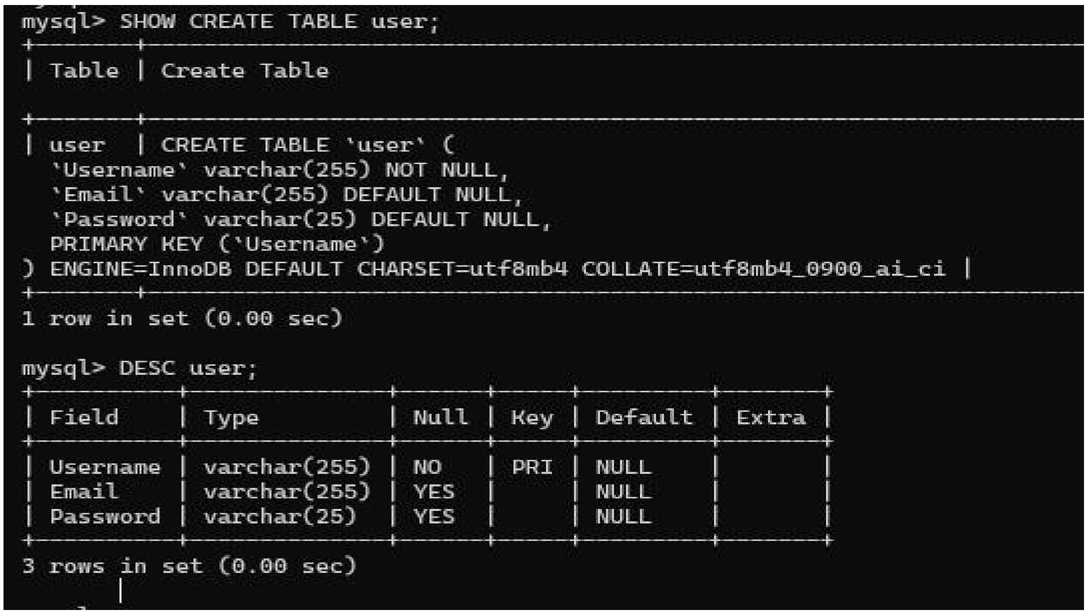
```
mysql> SHOW CREATE TABLE famous_food_images;
+--------------------+----------------------------------------------------------------------------------------------
----------------------------------------------------+
| Table              | Create Table

          |
+--------------------+----------------------------------------------------------------------------------------------
----------------------------------------------------+
| famous_food_images | CREATE TABLE `famous_food_images` (
  `food_id` int NOT NULL,
  `Image` varchar(255) NOT NULL,
  PRIMARY KEY (`food_id`,`Image`),
  CONSTRAINT `famous_food_images_ibfk_1` FOREIGN KEY (`food_id`) REFERENCES `famous_food` (`food_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+--------------------+----------------------------------------------------------------------------------------------
----------------------------------------------------+
1 row in set (0.00 sec)

mysql> DESC famous_food_images;
+---------+--------------+------+-----+---------+-------+
| Field   | Type         | Null | Key | Default | Extra |
+---------+--------------+------+-----+---------+-------+
| food_id | int          | NO   | PRI | NULL    |       |
| Image   | varchar(255) | NO   | PRI | NULL    |       |
+---------+--------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Organisation Table



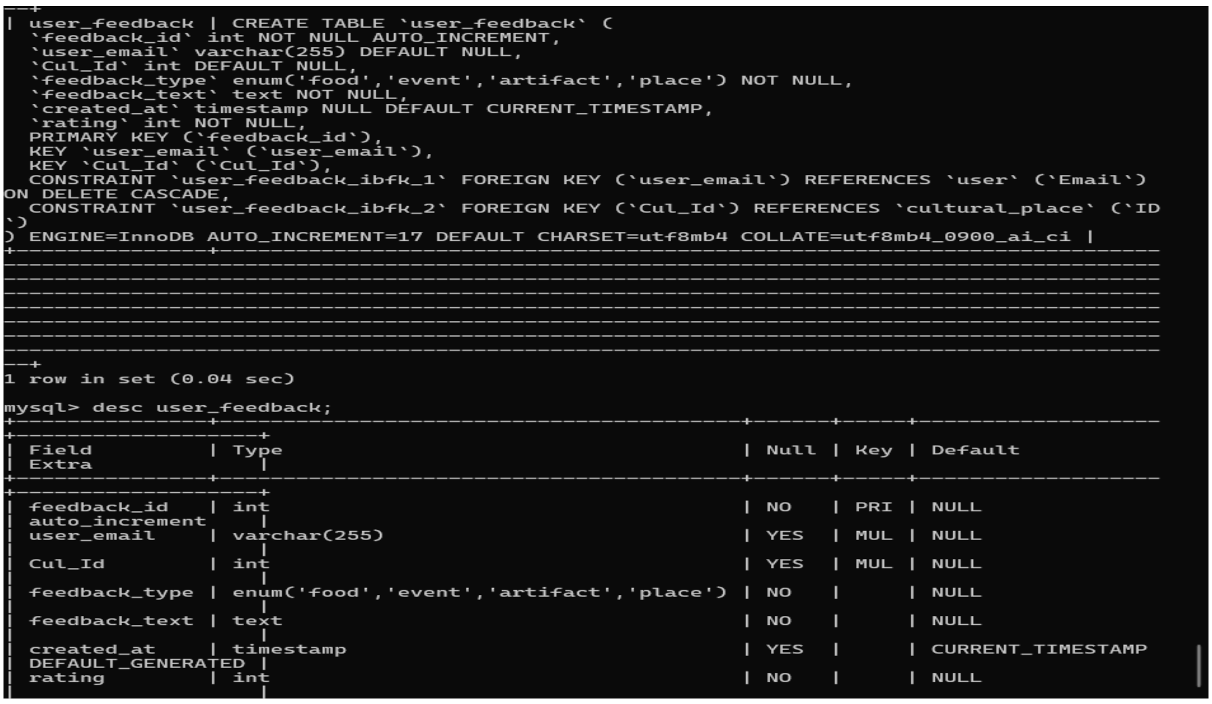
```
mysql> SHOW CREATE TABLE organisation;
+--------------+----------------------------------------------------------------------------------
| Table        | Create Table
+--------------+----------------------------------------------------------------------------------
| organisation | CREATE TABLE `organisation` (
  `Org_Id` int NOT NULL,
  `Name` varchar(255) NOT NULL,
  `Description` text,
  `City` varchar(255) DEFAULT NULL,
  `State` varchar(255) DEFAULT NULL,
  `Pincode` varchar(15) DEFAULT NULL,
  `Website_link` text,
  `Cul_id` int DEFAULT NULL,
  PRIMARY KEY (`Org_Id`),
  UNIQUE KEY `Name` (`Name`),
  KEY `Cul_id` (`Cul_id`),
  CONSTRAINT `organisation_ibfk_1` FOREIGN KEY (`Cul_id`) REFERENCES `cultural_place` (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+--------------+----------------------------------------------------------------------------------
1 row in set (0.00 sec)

mysql> DESC organisation;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| Org_Id       | int          | NO   | PRI | NULL    |       |
| Name         | varchar(255) | NO   | UNI | NULL    |       |
| Description  | text         | YES  |     | NULL    |       |
| City         | varchar(255) | YES  |     | NULL    |       |
| State        | varchar(255) | YES  |     | NULL    |       |
| Pincode      | varchar(15)  | YES  |     | NULL    |       |
| Website_link | text         | YES  |     | NULL    |       |
| Cul_id       | int          | YES  | MUL | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

User Table



```
mysql> SHOW CREATE TABLE user;
+-------+--------------------------------------------------------------------------
| Table | Create Table
+-------+--------------------------------------------------------------------------
| user  | CREATE TABLE `user` (
  `Username` varchar(255) NOT NULL,
  `Email` varchar(255) DEFAULT NULL,
  `Password` varchar(25) DEFAULT NULL,
  PRIMARY KEY (`Username`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-------+--------------------------------------------------------------------------
1 row in set (0.00 sec)

mysql> DESC user;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| Username | varchar(255) | NO   | PRI | NULL    |       |
| Email    | varchar(255) | YES  |     | NULL    |       |
| Password | varchar(25)  | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

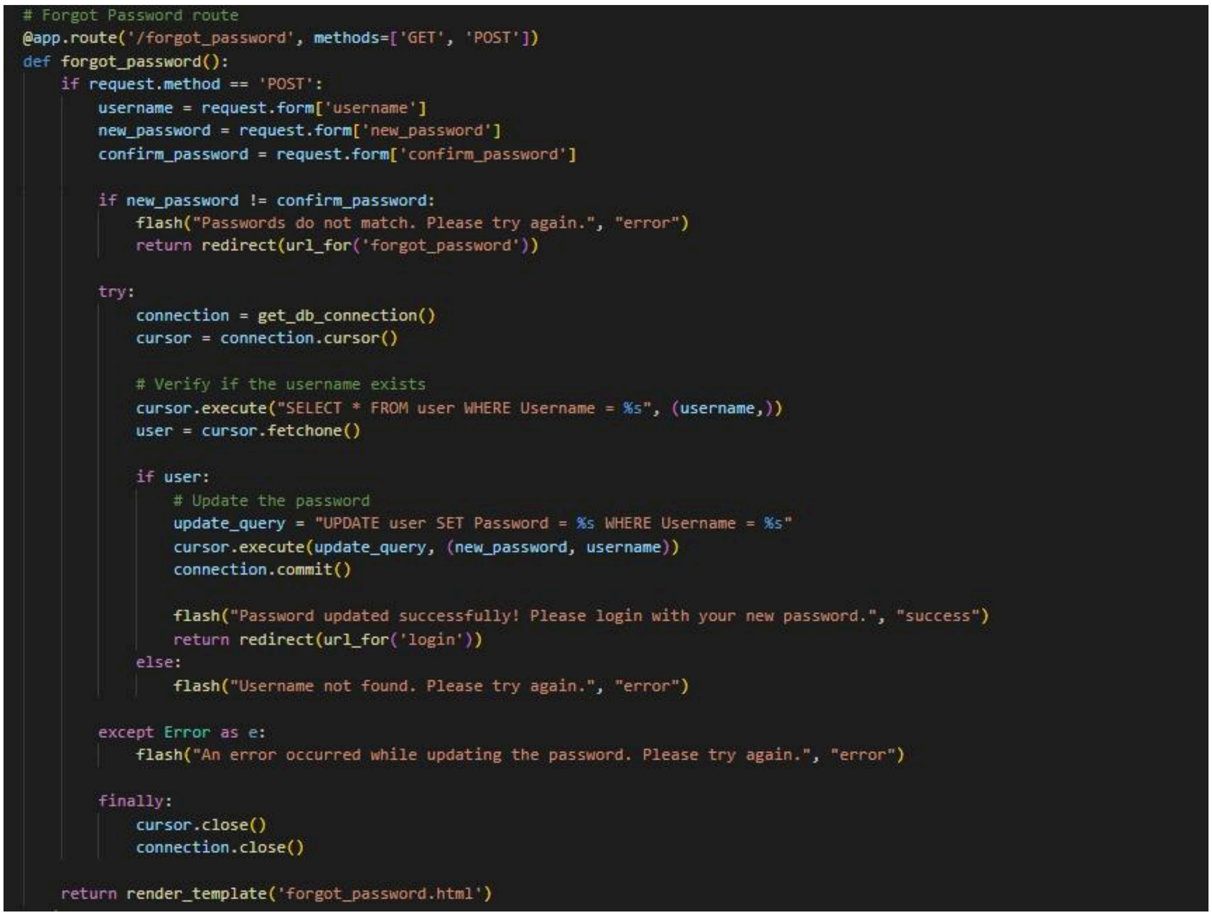
User feedback Table



```
--+
| user_feedback | CREATE TABLE `user_feedback` (
  `feedback_id` int NOT NULL AUTO_INCREMENT,
  `user_email` varchar(255) DEFAULT NULL,
  `Cul_Id` int DEFAULT NULL,
  `feedback_type` enum('food','event','artifact','place') NOT NULL,
  `feedback_text` text NOT NULL,
  `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `rating` int NOT NULL,
  PRIMARY KEY (`feedback_id`),
  KEY `user_email` (`user_email`),
  KEY `Cul_Id` (`Cul_Id`),
  CONSTRAINT `user_feedback_ibfk_1` FOREIGN KEY (`user_email`) REFERENCES `user` (`Email`)
ON DELETE CASCADE,
  CONSTRAINT `user_feedback_ibfk_2` FOREIGN KEY (`Cul_Id`) REFERENCES `cultural_place` (`ID
`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+----------------+------------------------------------------------------------------------
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
--+
1 row in set (0.04 sec)

mysql> desc user_feedback;
+----------------+-----------------------------------------+------+-----+-------------------
+-------------------+
| Field          | Type                                    | Null | Key | Default
| Extra             |
+----------------+-----------------------------------------+------+-----+-------------------
+-------------------+
| feedback_id    | int                                     | NO   | PRI | NULL
| auto_increment    |
| user_email     | varchar(255)                            | YES  | MUL | NULL
|                   |
| Cul_Id         | int                                     | YES  | MUL | NULL
|                   |
| feedback_type  | enum('food','event','artifact','place') | NO   |     | NULL
|                   |
| feedback_text  | text                                    | NO   |     | NULL
|                   |
| created_at     | timestamp                               | YES  |     | CURRENT_TIMESTAMP
| DEFAULT_GENERATED |
| rating         | int                                     | NO   |     | NULL
|                   |
```

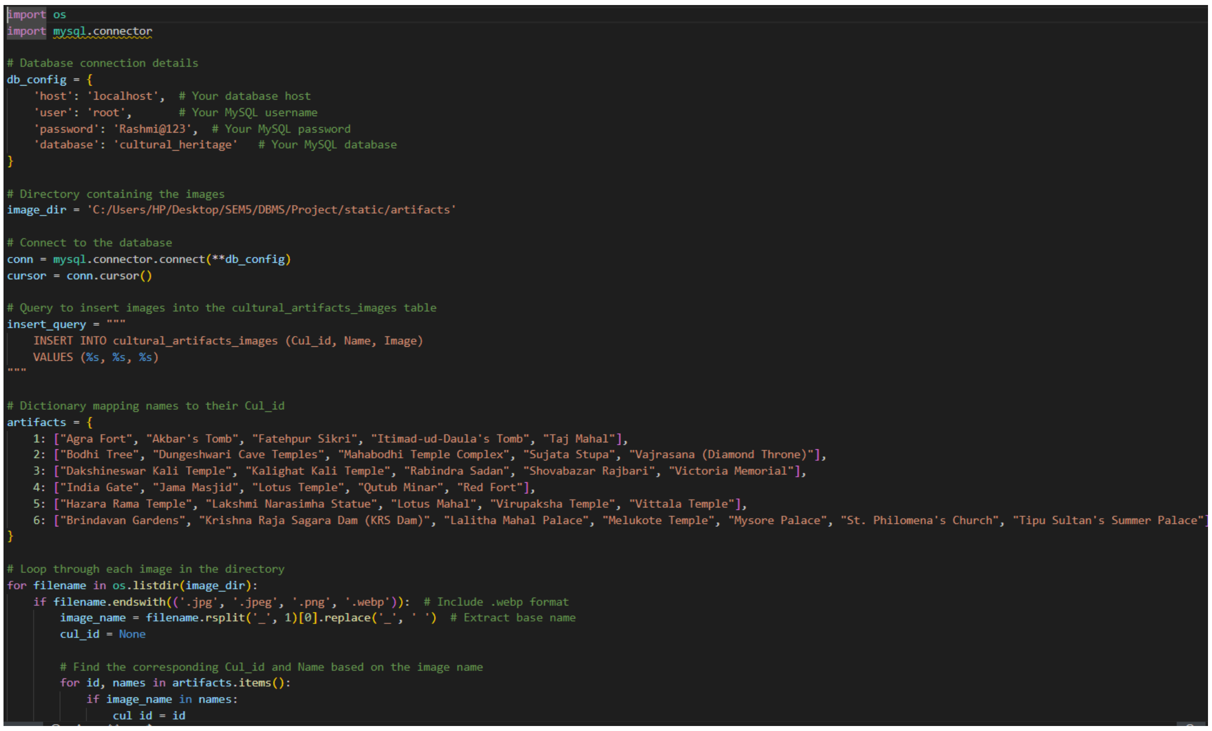
Updation:

SQL Query for updating the forget password

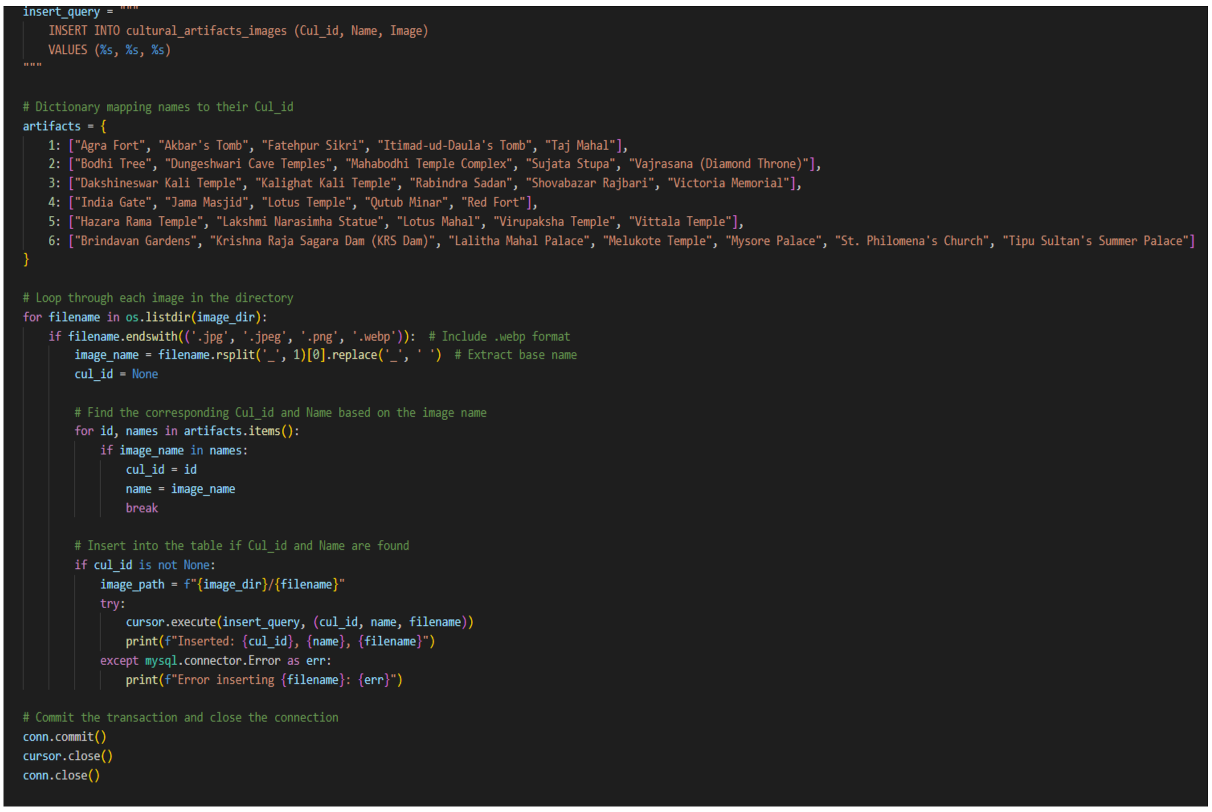


```python
# Forgot Password route
@app.route('/forgot_password', methods=['GET', 'POST'])
def forgot_password():
    if request.method == 'POST':
        username = request.form['username']
        new_password = request.form['new_password']
        confirm_password = request.form['confirm_password']

        if new_password != confirm_password:
            flash("Passwords do not match. Please try again.", "error")
            return redirect(url_for('forgot_password'))

        try:
            connection = get_db_connection()
            cursor = connection.cursor()

            # Verify if the username exists
            cursor.execute("SELECT * FROM user WHERE Username = %s", (username,))
            user = cursor.fetchone()

            if user:
                # Update the password
                update_query = "UPDATE user SET Password = %s WHERE Username = %s"
                cursor.execute(update_query, (new_password, username))
                connection.commit()

                flash("Password updated successfully! Please login with your new password.", "success")
                return redirect(url_for('login'))
            else:
                flash("Username not found. Please try again.", "error")

        except Error as e:
            flash("An error occurred while updating the password. Please try again.", "error")

        finally:
            cursor.close()
            connection.close()

    return render_template('forgot_password.html')
```

Python Program to insert the images into the Artifact Image table



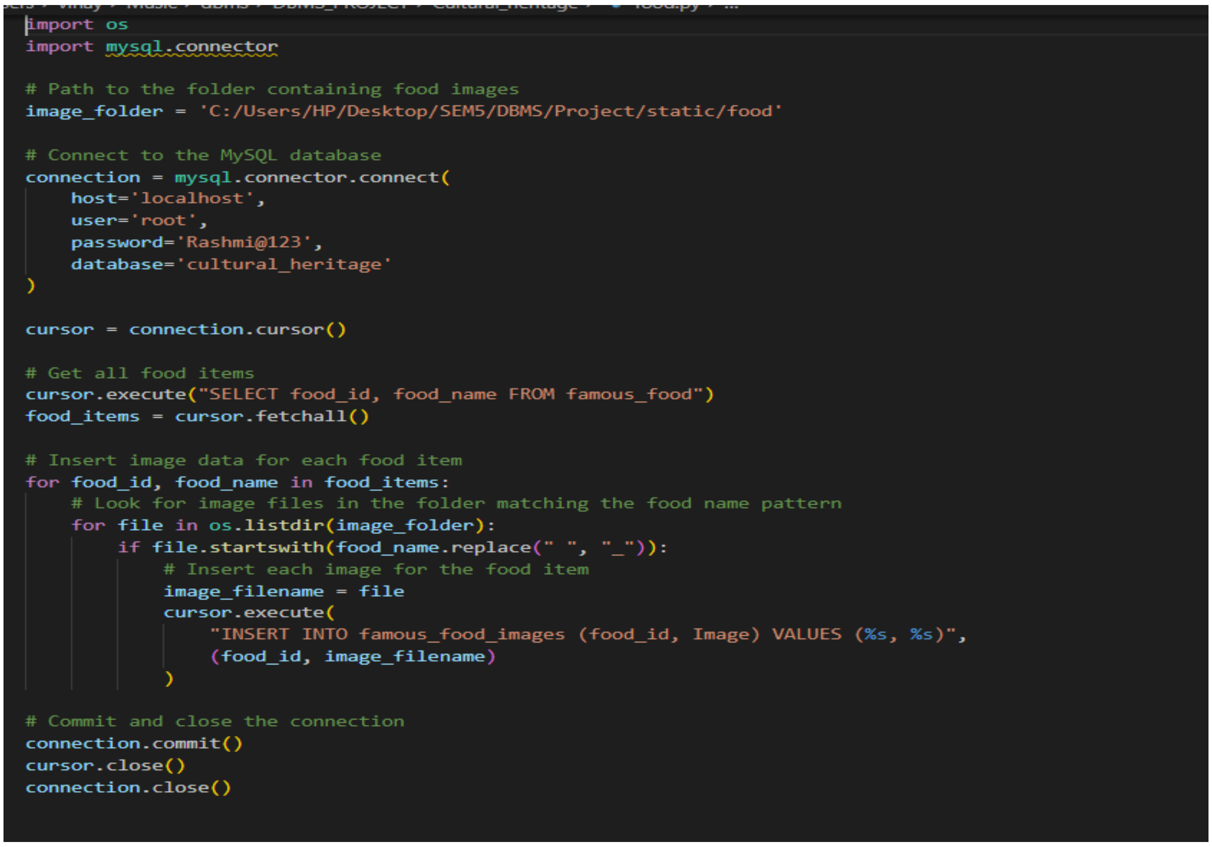
```python
import os
import mysql.connector

# Database connection details
db_config = {
    'host': 'localhost',  # Your database host
    'user': 'root',       # Your MySQL username
    'password': 'Rashmi@123',  # Your MySQL password
    'database': 'cultural_heritage'   # Your MySQL database
}

# Directory containing the images
image_dir = 'C:/Users/HP/Desktop/SEM5/DBMS/Project/static/artifacts'

# Connect to the database
conn = mysql.connector.connect(**db_config)
cursor = conn.cursor()

# Query to insert images into the cultural_artifacts_images table
insert_query = """
    INSERT INTO cultural_artifacts_images (Cul_id, Name, Image)
    VALUES (%s, %s, %s)
"""

# Dictionary mapping names to their Cul_id
artifacts = {
    1: ["Agra Fort", "Akbar's Tomb", "Fatehpur Sikri", "Itimad-ud-Daula's Tomb", "Taj Mahal"],
    2: ["Bodhi Tree", "Dungeshwari Cave Temples", "Mahabodhi Temple Complex", "Sujata Stupa", "Vajrasana (Diamond Throne)"],
    3: ["Dakshineswar Kali Temple", "Kalighat Kali Temple", "Rabindra Sadan", "Shovabazar Rajbari", "Victoria Memorial"],
    4: ["India Gate", "Jama Masjid", "Lotus Temple", "Qutub Minar", "Red Fort"],
    5: ["Hazara Rama Temple", "Lakshmi Narasimha Statue", "Lotus Mahal", "Virupaksha Temple", "Vittala Temple"],
    6: ["Brindavan Gardens", "Krishna Raja Sagara Dam (KRS Dam)", "Lalitha Mahal Palace", "Melukote Temple", "Mysore Palace", "St. Philomena's Church", "Tipu Sultan's Summer Palace"]
}

# Loop through each image in the directory
for filename in os.listdir(image_dir):
    if filename.endswith(('.jpg', '.jpeg', '.png', '.webp')):  # Include .webp format
        image_name = filename.rsplit('_', 1)[0].replace('_', ' ')  # Extract base name
        cul_id = None

        # Find the corresponding Cul_id and Name based on the image name
        for id, names in artifacts.items():
            if image_name in names:
                cul_id = id
```



```python
insert_query = """
    INSERT INTO cultural_artifacts_images (Cul_id, Name, Image)
    VALUES (%s, %s, %s)
"""

# Dictionary mapping names to their Cul_id
artifacts = {
    1: ["Agra Fort", "Akbar's Tomb", "Fatehpur Sikri", "Itimad-ud-Daula's Tomb", "Taj Mahal"],
    2: ["Bodhi Tree", "Dungeshwari Cave Temples", "Mahabodhi Temple Complex", "Sujata Stupa", "Vajrasana (Diamond Throne)"],
    3: ["Dakshineswar Kali Temple", "Kalighat Kali Temple", "Rabindra Sadan", "Shovabazar Rajbari", "Victoria Memorial"],
    4: ["India Gate", "Jama Masjid", "Lotus Temple", "Qutub Minar", "Red Fort"],
    5: ["Hazara Rama Temple", "Lakshmi Narasimha Statue", "Lotus Mahal", "Virupaksha Temple", "Vittala Temple"],
    6: ["Brindavan Gardens", "Krishna Raja Sagara Dam (KRS Dam)", "Lalitha Mahal Palace", "Melukote Temple", "Mysore Palace", "St. Philomena's Church", "Tipu Sultan's Summer Palace"]
}

# Loop through each image in the directory
for filename in os.listdir(image_dir):
    if filename.endswith(('.jpg', '.jpeg', '.png', '.webp')):  # Include .webp format
        image_name = filename.rsplit('_', 1)[0].replace('_', ' ')  # Extract base name
        cul_id = None

        # Find the corresponding Cul_id and Name based on the image name
        for id, names in artifacts.items():
            if image_name in names:
                cul_id = id
```

```python
                name = image_name
                break

        # Insert into the table if Cul_id and Name are found
        if cul_id is not None:
            image_path = f"{image_dir}/{filename}"
            try:
                cursor.execute(insert_query, (cul_id, name, filename))
                print(f"Inserted: {cul_id}, {name}, {filename}")
            except mysql.connector.Error as err:
                print(f"Error inserting {filename}: {err}")

# Commit the transaction and close the connection
conn.commit()
cursor.close()
conn.close()
```

Python Program to insert the images into the Famous Food table

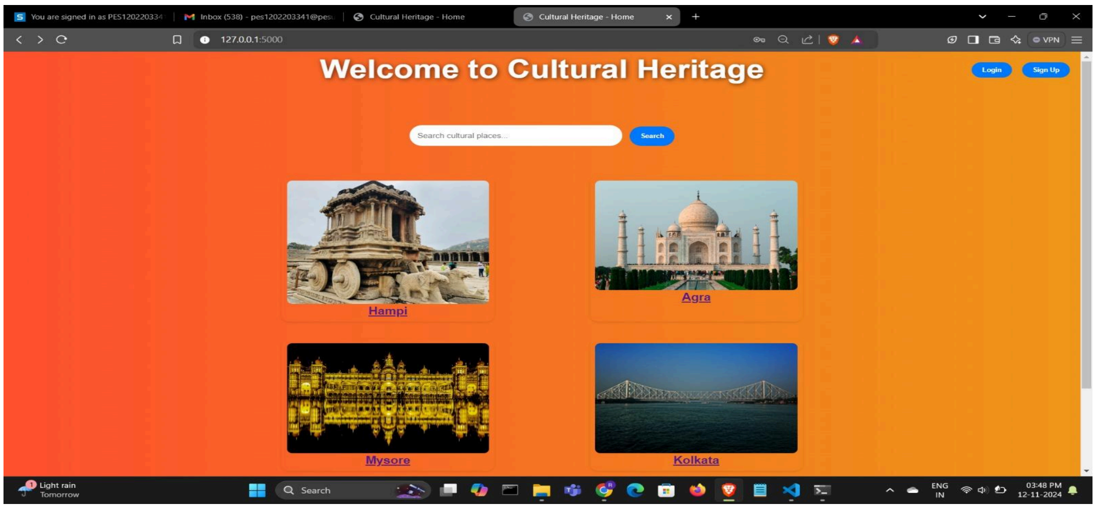


```python
import os
import mysql.connector

# Path to the folder containing food images
image_folder = 'C:/Users/HP/Desktop/SEM5/DBMS/Project/static/food'

# Connect to the MySQL database
connection = mysql.connector.connect(
    host='localhost',
    user='root',
    password='Rashmi@123',
    database='cultural_heritage'
)

cursor = connection.cursor()

# Get all food items
cursor.execute("SELECT food_id, food_name FROM famous_food")
food_items = cursor.fetchall()

# Insert image data for each food item
for food_id, food_name in food_items:
    # Look for image files in the folder matching the food name pattern
    for file in os.listdir(image_folder):
        if file.startswith(food_name.replace(" ", "_")):
            # Insert each image for the food item
            image_filename = file
            cursor.execute(
                "INSERT INTO famous_food_images (food_id, Image) VALUES (%s, %s)",
                (food_id, image_filename)
            )

# Commit and close the connection
connection.commit()
cursor.close()
connection.close()
```

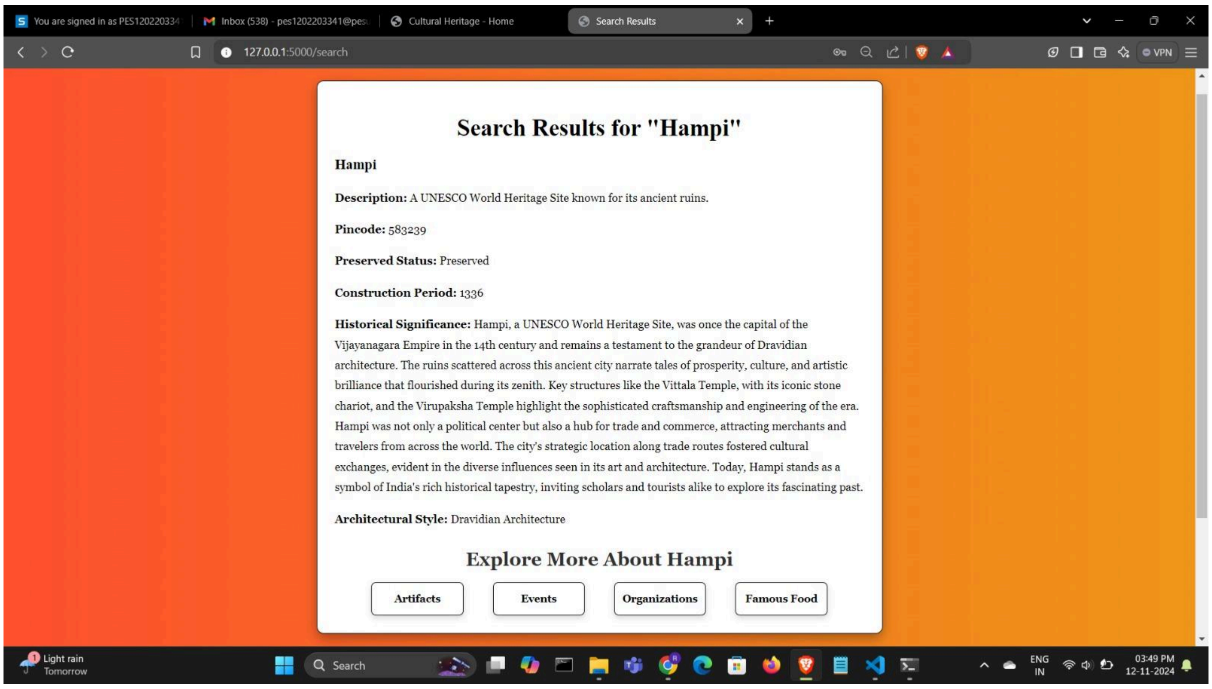
<u>UI of cultural heritage website</u>:

- Handles the "forgot password" functionality.
- Retrieves username and new password from the user.
- Verifies password and updates it in the database.
- Provides feedback to the user.
- Automates the process of adding images to a database.
- Associates images with specific items (artifacts or food) based on their filenames.
- Stores image filenames, item IDs, and item names in a database table.
- Uses database connection, file handling, and SQL queries to achieve this.

Home Page



Search Result Page



Search Results for "Hampi"

Hampi

Description: A UNESCO World Heritage Site known for its ancient ruins.

Pincode: 583239

Preserved Status: Preserved

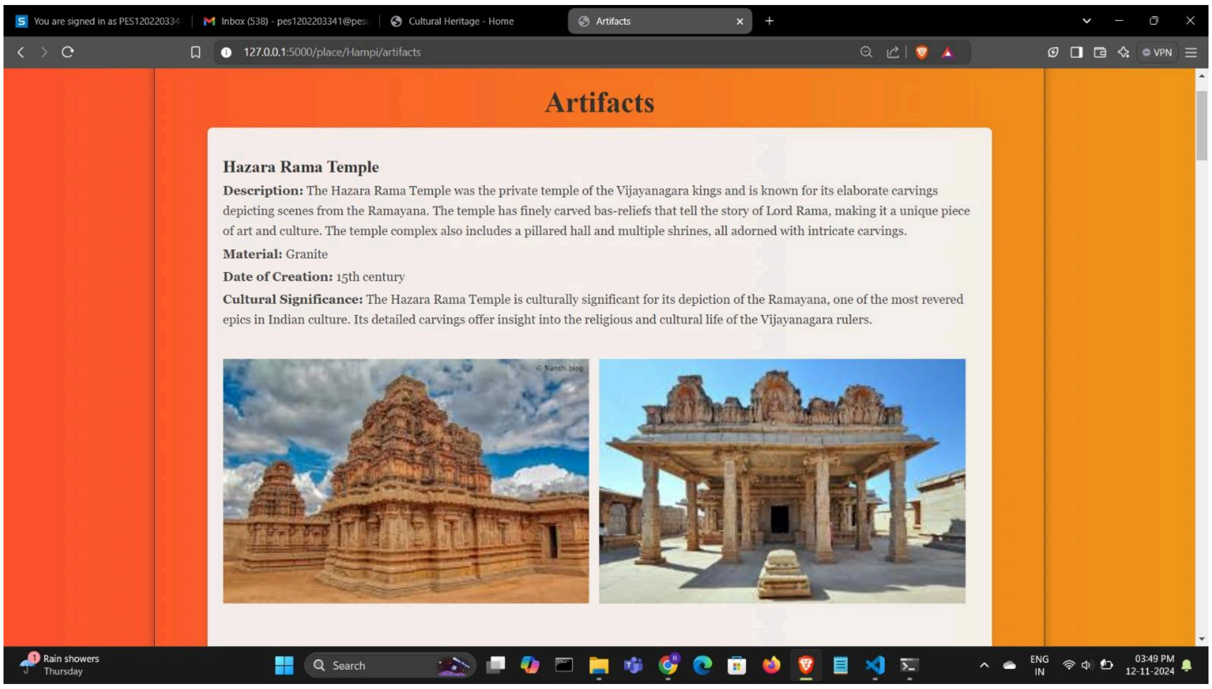
Construction Period: 1336

Historical Significance: Hampi, a UNESCO World Heritage Site, was once the capital of the Vijayanagara Empire in the 14th century and remains a testament to the grandeur of Dravidian architecture. The ruins scattered across this ancient city narrate tales of prosperity, culture, and artistic brilliance that flourished during its zenith. Key structures like the Vittala Temple, with its iconic stone chariot, and the Virupaksha Temple highlight the sophisticated craftsmanship and engineering of the era. Hampi was not only a political center but also a hub for trade and commerce, attracting merchants and travelers from across the world. The city's strategic location along trade routes fostered cultural exchanges, evident in the diverse influences seen in its art and architecture. Today, Hampi stands as a symbol of India's rich historical tapestry, inviting scholars and tourists alike to explore its fascinating past.

Architectural Style: Dravidian Architecture

Explore More About Hampi

Artifact Page



Artifacts

Hazara Rama Temple

Description: The Hazara Rama Temple was the private temple of the Vijayanagara kings and is known for its elaborate carvings depicting scenes from the Ramayana. The temple has finely carved bas-reliefs that tell the story of Lord Rama, making it a unique piece of art and culture. The temple complex also includes a pillared hall and multiple shrines, all adorned with intricate carvings.

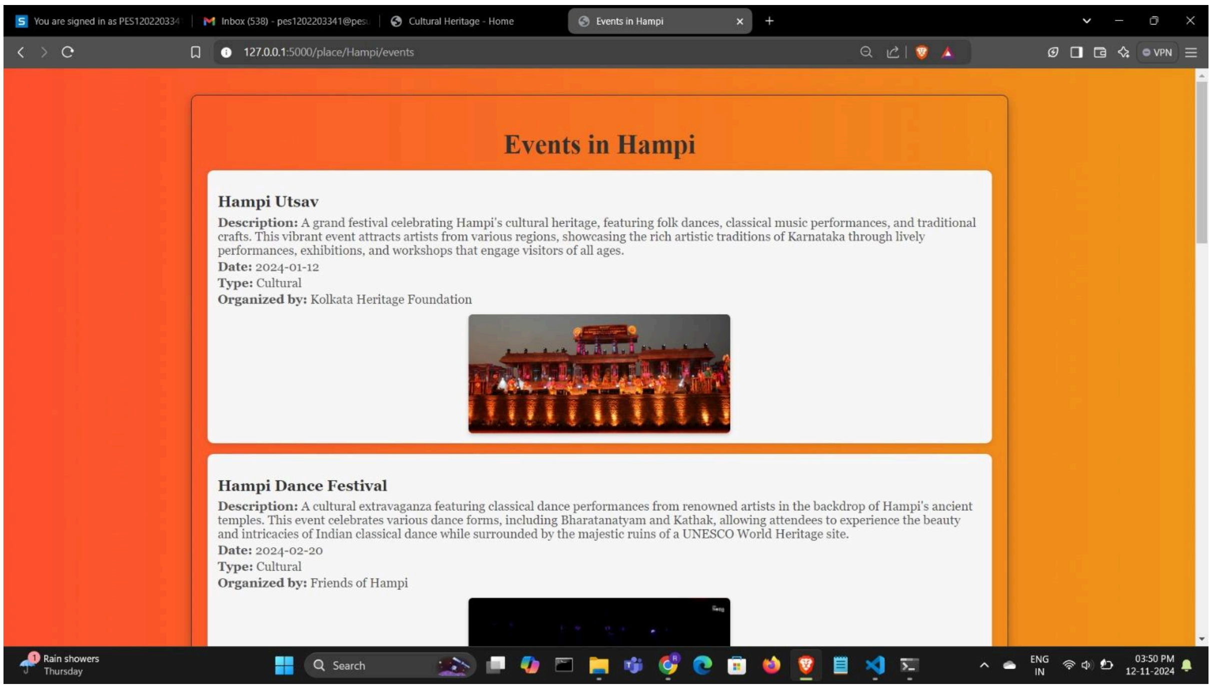
Material: Granite

Date of Creation: 15th century

Cultural Significance: The Hazara Rama Temple is culturally significant for its depiction of the Ramayana, one of the most revered epics in Indian culture. Its detailed carvings offer insight into the religious and cultural life of the Vijayanagara rulers.

You are signed in as PES120220334
Inbox (538) - pes1202203341@pes
Cultural Heritage - Home
Artifacts
127.0.0.1:5000/place/Hampi/artifacts
Rain showers
Thursday
Search
03:49 PM
12-11-2024

Events Page



127.0.0.1:5000/place/Hampi/events

Events in Hampi

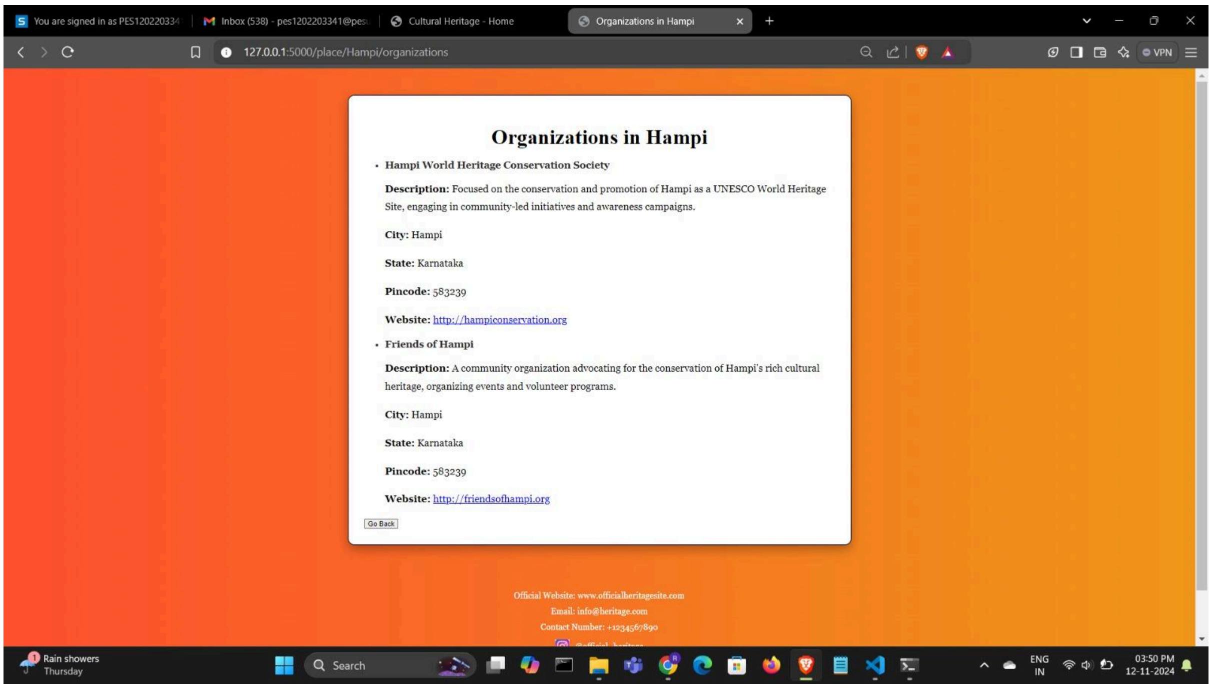
Hampi Utsav

Description: A grand festival celebrating Hampi's cultural heritage, featuring folk dances, classical music performances, and traditional crafts. This vibrant event attracts artists from various regions, showcasing the rich artistic traditions of Karnataka through lively performances, exhibitions, and workshops that engage visitors of all ages.
Date: 2024-01-12
Type: Cultural
Organized by: Kolkata Heritage Foundation

Hampi Dance Festival

Description: A cultural extravaganza featuring classical dance performances from renowned artists in the backdrop of Hampi's ancient temples. This event celebrates various dance forms, including Bharatanatyam and Kathak, allowing attendees to experience the beauty and intricacies of Indian classical dance while surrounded by the majestic ruins of a UNESCO World Heritage site.
Date: 2024-02-20
Type: Cultural
Organized by: Friends of Hampi

Organisation Page



127.0.0.1:5000/place/Hampi/organizations

Organizations in Hampi

- **Hampi World Heritage Conservation Society**

  Description: Focused on the conservation and promotion of Hampi as a UNESCO World Heritage Site, engaging in community-led initiatives and awareness campaigns.

  City: Hampi

  State: Karnataka

  Pincode: 583239

  Website: http://hampiconservation.org

- **Friends of Hampi**

  Description: A community organization advocating for the conservation of Hampi's rich cultural heritage, organizing events and volunteer programs.

  City: Hampi

  State: Karnataka

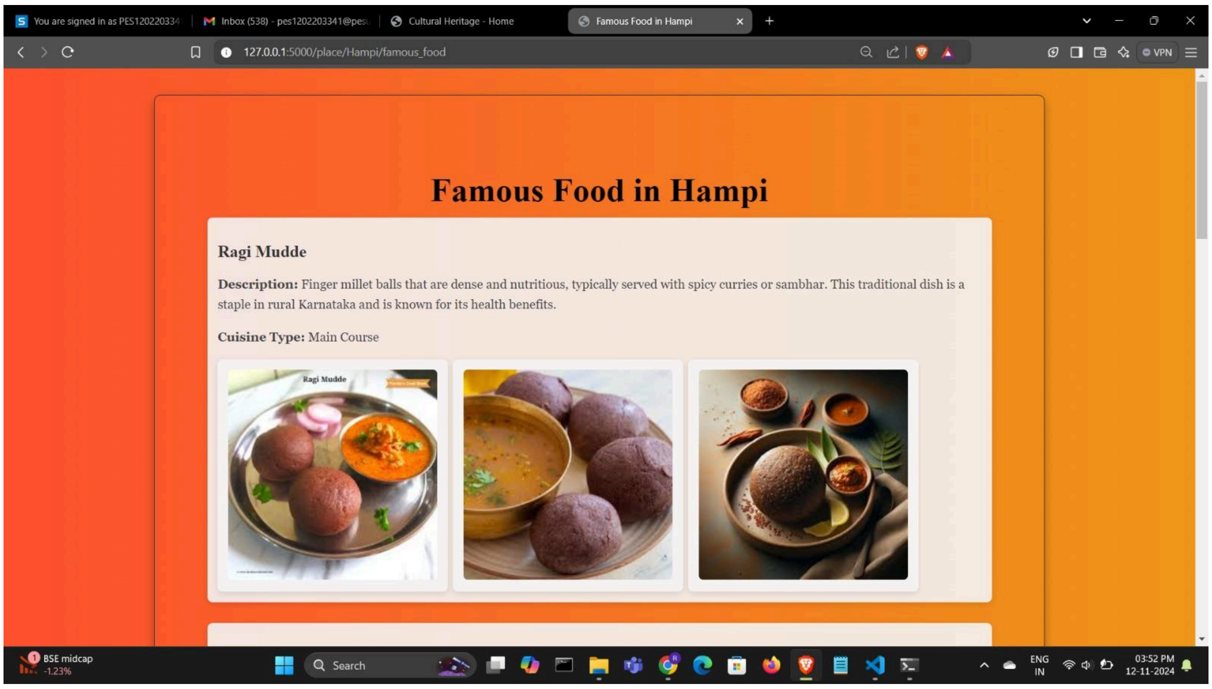
  Pincode: 583239

  Website: http://friendsofhampi.org

Go Back

Official Website: www.officialheritagesite.com
Email: info@heritage.com
Contact Number: +1234567890

Famous Food Page



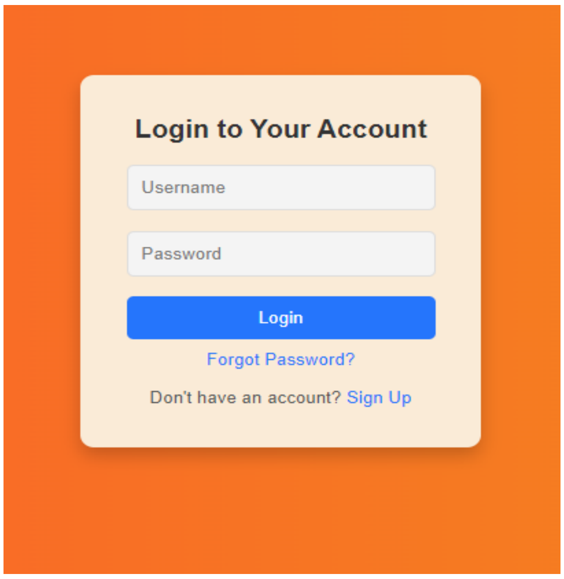
Famous Food in Hampi

Ragi Mudde

Description: Finger millet balls that are dense and nutritious, typically served with spicy curries or sambhar. This traditional dish is a staple in rural Karnataka and is known for its health benefits.

Cuisine Type: Main Course

Login Page



Login to Your Account

Username

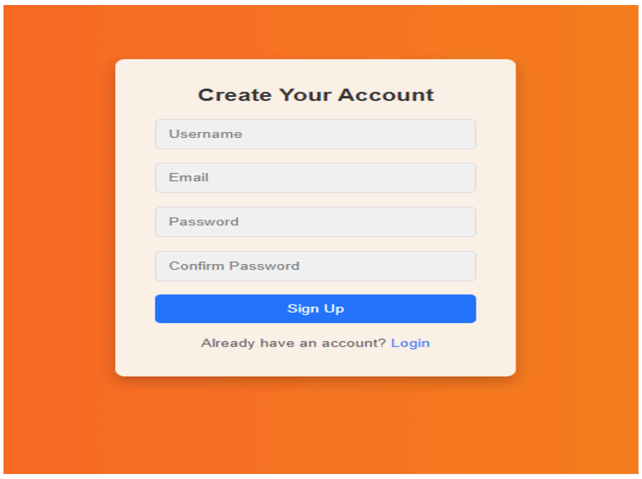
Password

Login

Forgot Password?

Don't have an account? Sign Up

Signup page



Create Your Account

Username

Email

Password

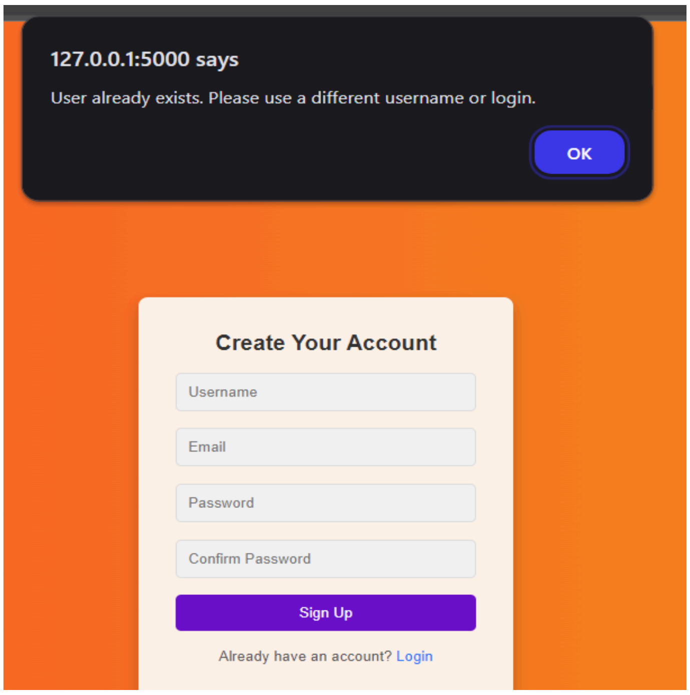
Confirm Password

Sign Up

Already have an account? Login

Trigger Execution On Web Page:

Execution of trigger on the web page when the username is matched with already existing username during signup

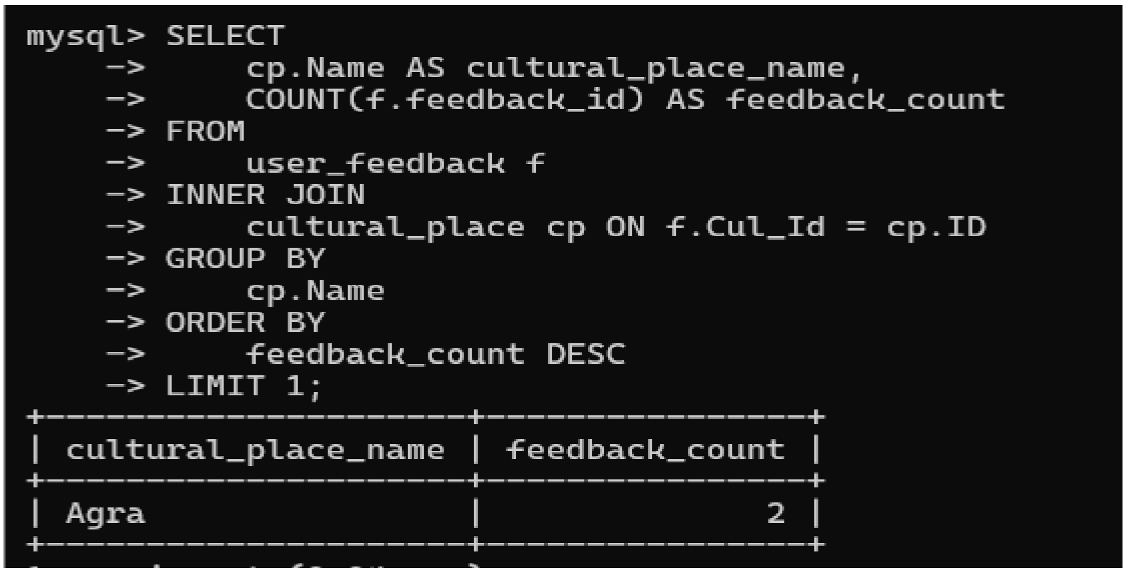


127.0.0.1:5000 says

User already exists. Please use a different username or login.

Aggregate function

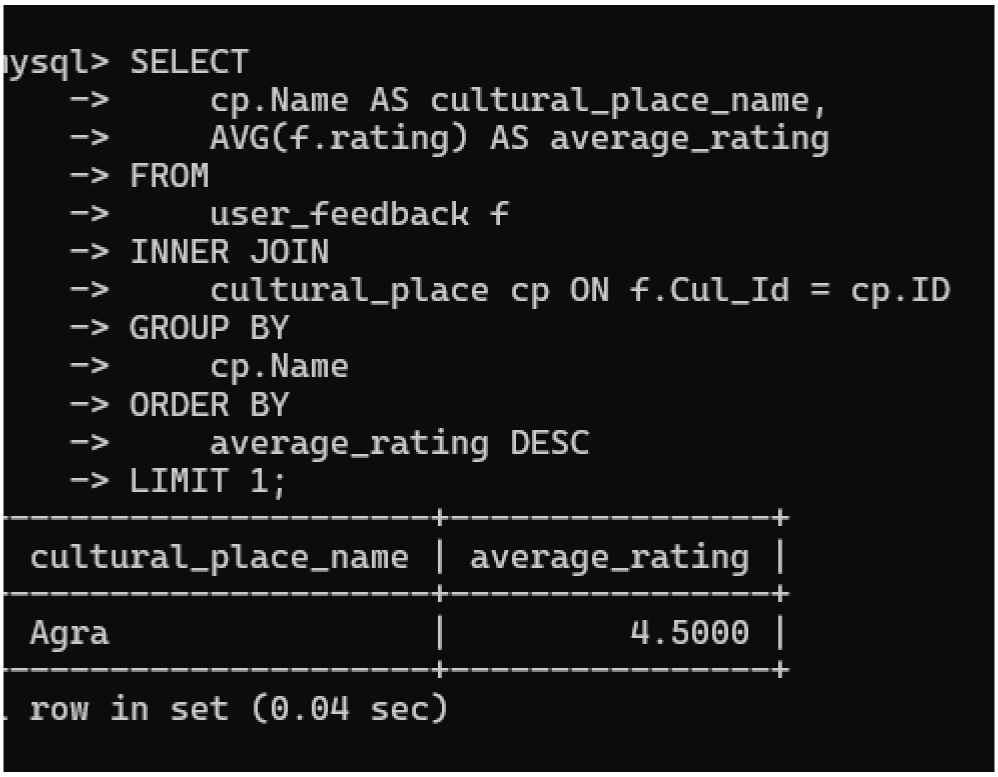
Popular Cultural Places:
1.Using feedback count



```
mysql> SELECT
    ->     cp.Name AS cultural_place_name,
    ->     COUNT(f.feedback_id) AS feedback_count
    -> FROM
    ->     user_feedback f
    -> INNER JOIN
    ->     cultural_place cp ON f.Cul_Id = cp.ID
    -> GROUP BY
    ->     cp.Name
    -> ORDER BY
    ->     feedback_count DESC
    -> LIMIT 1;
+---------------------+----------------+
| cultural_place_name | feedback_count |
+---------------------+----------------+
| Agra                |              2 |
+---------------------+----------------+
```

This query is used to find the cultural place with the highest number of feedback entries Identifying

2)On average rating:

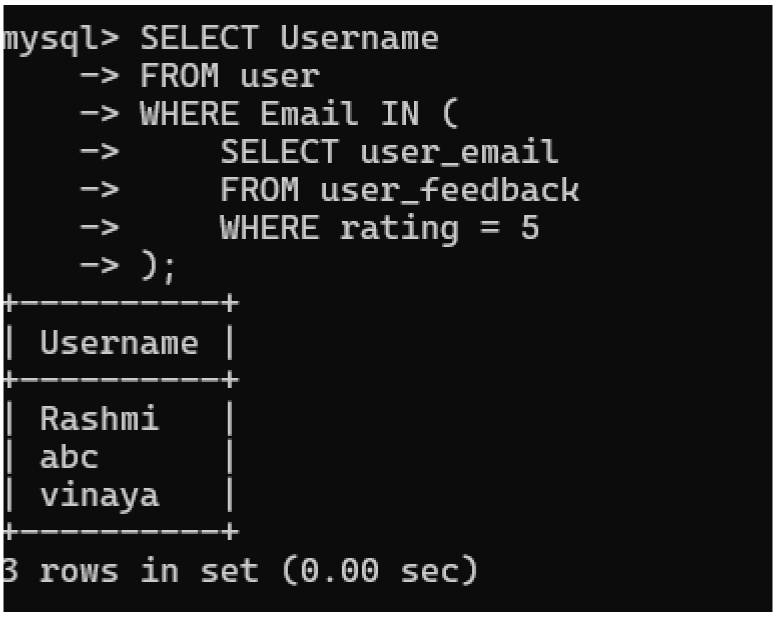


```
ysql> SELECT
    ->     cp.Name AS cultural_place_name,
    ->     AVG(f.rating) AS average_rating
    -> FROM
    ->     user_feedback f
    -> INNER JOIN
    ->     cultural_place cp ON f.Cul_Id = cp.ID
    -> GROUP BY
    ->     cp.Name
    -> ORDER BY
    ->     average_rating DESC
    -> LIMIT 1;
---------------------+----------------+
 cultural_place_name | average_rating |
---------------------+----------------+
 Agra                |         4.5000 |
---------------------+----------------+
 row in set (0.04 sec)
```

This query is used to find the cultural place with the highest average user rating in the user_feedback

Nested queries:

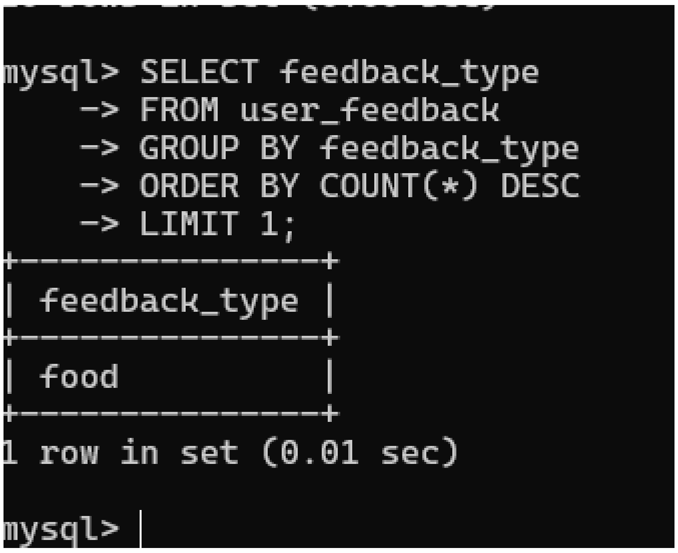
1) Retrieve Usernames Who Gave Maximum Rating (5)



```
mysql> SELECT Username
    -> FROM user
    -> WHERE Email IN (
    ->     SELECT user_email
    ->     FROM user_feedback
    ->     WHERE rating = 5
    -> );
+----------+
| Username |
+----------+
| Rashmi   |
| abc      |
| vinaya   |
+----------+
3 rows in set (0.00 sec)
```

Find usernames who have given a rating of 5 in any feedback:

2)Get Most Frequently Received Feedback Type



```
mysql> SELECT feedback_type
    -> FROM user_feedback
    -> GROUP BY feedback_type
    -> ORDER BY COUNT(*) DESC
    -> LIMIT 1;
+---------------+
| feedback_type |
+---------------+
| food          |
+---------------+
1 row in set (0.01 sec)

mysql>
```

Retrieve the feedback type that occurs most frequently across all feedback: